Low Cost Power Saving UV Light Exposure Box for PCB Developing

by

Beniedick I. Hernandez
Juan Manuel B. Ramos
Arman A. Taylo

A Design Report Submitted to the School of Electrical Engineering, Electronics Engineering, and Computer Engineering in Partial Fulfillment of the Requirements for the Degree

Bachelor of Science in Computer Engineering

Mapua Institute of Technology
February 2011

Approval Sheet

Mapua Institute of Technology
School of EECE

This is to certify that we have supervised the preparation of and read the design report prepared by **Beniedick I. Hernandez, Juan Manuel B. Ramos** and **Arman A. Taylo** entitled ***LOW COST POWER SAVING UV LIGHT EXPOSURE BOX FOR PCB DEVELOPING*** and that the said report has been submitted for final examination by the Oral Examination Committee.

Engr. Analyn Niere-Yumang
Design Adviser

As members of the Oral Examination Committee, we certify that we have examined this design report, presented before the committee on **March 5, 2011**, and hereby recommended that it be accepted as fulfillment of the design requirement for the degree in **Bachelor of Science in Computer Engineering.**

Engr. Emma Ruth M. Tiong
Panel Member

Engr. Marianne M. Sejera
Panel Member

Engr. Joshua Cuesta
Panel Member

This design report is hereby approved and accepted by the School of Electrical Engineering, Electronics Engineering, and Computer Engineering in partial fulfillment of the requirements for the degree in **Bachelor of Science in Computer Engineering.**

Dr. Felicito S. Caluyo
Dean, School of EECE

ACKNOWLEDGEMENT

The group would like to give their sincerest gratitude to the following people for their extensive support and guidance they made in this Computer Engineering Design:

First and foremost to our Almighty Father, who made all the things possible, we thank Him for giving us the strength and wisdom to finish this design.

To our parents, we thank them for their moral support and encouragement as we go through the struggles of making this design possible.

To our professor, Prof. Noel B. Linsangan, we give our great appreciation for letting the group work on this design and for his perseverance in bringing out the best in every member of the group. We also thank him for his considerations and concern for the group.

To our adviser, Engr. Analyn Niere-Yumang, we thank her for giving us enough time and unfailing support throughout the whole design construction, especially on the documentation.

Beniedick I. Hernandez

Juan Manuel B.Ramos

Arman A. Taylo

ROLES AND RESPONSIBILITIES OF GROUP MEMBERS

Each member contributed on researching the given design project and the development of the prototype.

The following shows the list of responsibilities for each member:

Beniedick I. Hernandez

- Circuit Design
- Hardware Design
- Software Design
- Documentation

Juan Manuel B.Ramos

- Circuit Design
- Hardware Design
- Software Design
- Documentation

Arman A. Taylo

- Circuit Design
- Hardware Design
- Software Design
- Documentation

TABLE OF CONTENTS

TITLE PAGE i

APPROVAL SHEET **Error! Bookmark not defined.**

ACKNOWLEDGEMENT ii

ROLES AND RESPONSIBILITIES OF GROUP MEMBERS iv

TABLE OF CONTENTS v

LIST OF TABLES vii

LIST OF FIGURES viii

ABSTRACT ix

CHAPTER 1: DESIGN BACKGROUND AND INTRODUCTION 1

- Background 1
- Statement of the Problem 2
- Objective of the Design 3
- Significance of the Design 4
- Scope and Delimitations 5
- Definition of Terms 6

CHAPTER 2: REVIEW OF RELATED DESIGN LITERATURES AND STUDIES 8

CHAPTER 3: DESIGN PROCEDURES 16

- Hardware Development 20
- Block Diagram 21
- Schematic Diagram 24
- Software Development 27
- Prototype Development 28

CHAPTER 4: TESTING, PRESENTATION, AND INTERPRETATION OF DATA 33

- Ultraviolet LEDs as Light Source 33

Ultraviolet LEDs for Less Energy Consumption 34
Preset Exposure Time 37
CHAPTER 5: CONCLUSION AND RECOMMENDATION 40
CONCLUSION 40
RECOMMENDATION 40
BIBLIOGRAPHY 42
APPENDIX 43
APPENDIX A: OPERATION'S MANUAL 43
APPENDIX B: PICTURES OF PROTOTYPE 44
APPENDIX C: PROGRAM LISTING 45
APPENDIX D: DATASHEETS 83

LIST OF TABLES

Table 2.1 Lifetime of LightSource

Table 3.1 Decomposition of Components

Table 4.1 UV LEDs as Light Source

Table 4.2 UV Lamp PCB Exposure Box

Table 4.3 UV LED PCB Exposure Box

Table 4.4 Preset time for Acetate

Table 4.4 Preset time for Tracing Paper

LIST OF FIGURES

Figure 3.0 Design Methodology Flowchart

Figure 3.1 Conceptual Diagram

Figure 3.2 Block Diagram

Figure 3.2.1 User Input

Figure 3.2.2 Single-sided

Figure 3.2.3 Double-sided

Figure 3.2.4 Acetate

Figure 3.2.5 Tracing Paper

Figure 3.3 Schematic Diagram

Figure 3.3.1 Schematic Diagram

Figure 3.4 System Flowchart

Figure 3.5.1 LCD Idle State

Figure 3.5.2 LCD Confirmation Screen

Figure 3.5.3 Invalid Size

Figure 3.5.4 Type of Paper

Figure 3.5.5 Type of PCB

Figure 3.5.6 Timer

Figure 6.1 Prototype

Figure 6.2 Controller Box

Figure 6.3 LED Panels (TURNED OFF)

Figure 6.4 LED Panels (TURNED ON)

ABSTRACT

Fabricating Printed Circuit Board is almost an inescapable task all electronic constructors must somehow go through. The problem is that the process of fabrication of PCB can be a little pricey especially in double sided types. Thus, the purpose of the design project was to provide a device that can actually support the needs in fabricating PCB in a minimal cost but effective way. These led to the innovation of the existing UV exposure boxes. Using ultraviolet LEDs as an alternative for the ultraviolet lamps helps the design to lessen its overall cost and reduce energy consumption through exposing optimal dimensions of PCBs. Also, having the design's ability to assure the best time of exposure, and warn and turn itself off when the exposure time is finally over will surely help the fabricator. In general, the design can assure the quality of service that electronic constructors needs even in minimal cost.

Keywords: Printed Circuit Board, double-sided PCB, Ultraviolet, LED, exposure box

CHAPTER 1

DESIGN BACKGROUND AND INTRODUCTION

Background

Fabricating Printed Circuit Board PCB is almost an inescapable task all electronic constructors must somehow go through. The PCB is both important and essential part of the electronic circuit assembly. The final performance and reliability outcome of the circuit being built is dependent to a large extent how well the PCB was built. That being said, do it yourself PCB proficiency should rank among the most important skill every electronic circuit constructor must hone.

There are lots of methods that are floating around. Some uses permanent marker, masking tape, scotch tape, presensitized, toner transfer and even wax method. They all have a satisfying result depending on someone's actual need. If trace width is important, presensitized will beat almost all the method mentioned above. While it is more capable of high resolution PCB, there are lots of variables involved in achieving a good result, some of which are, the intensity of light to be used, duration of the exposure, quality of photo-resist material used, strength of the developer used and the duration of the actual developing process.

For these reasons UV exposure box became an extremely useful piece of kit in fabricating PCBs. The problem is that they can be a little pricey for the hobby enthusiast especially in double sided types.

Having to design a UV exposure box that will eliminate this problem will surely benefit the electronic constructors. Timer and buzzer are some additional features that to be added to the design to enhance its performance and functionality.

Thus, the purpose of the design project was to provide the electronic constructors a device that can actually support their needs in fabricating PCB in a minimal cost but effective way. These reasons led to the innovation of the existing UV exposure boxes.

Statement of the Problem

Fabrication of PCBs can be done in different sizes, depending on the specifications that the electronic constructor needs. Available designs of the UV exposure boxes were a little expensive and only uses UV fluorescent bulb which cannot be set on the optimal dimension of the PCBs to be exposed. Also, other UV exposure boxes only support single sided PCB causing a lot of time to expose both sides of the PCB. In addition, a timer and buzzer weren't considered as a necessity from the old design of the exposure boxes causing a lot of rough estimation in the exposure time of the PCB. These problems posed certain

inquiries that are necessary to be taken into account, to achieve the goal of this research.

One major concern that these problems highlighted is the high cost of UV exposure boxes. Given the limitations of the research, this problem posed the inquiry as to how it is possible to build a UV exposure box at a minimal cost. Another problem that arouse is the question as to how can the device help minimizing the energy consumption while in the process of PCB exposure. Other inquiries that concerns the performance of the device include questions as to how it can support double sided and single sided PCB, its ability to assure the best time of exposure, and how can the device warn the PCB fabricator when the exposure time is finally over.

Objective of the Design

The primary objective of the design was to create a low cost fully-featured UV exposure box.

Specifically this will include:

1. To lessen energy consumption, through exposing the optimal sizes of the PCB by using UV LEDs.
2. To support single and double sided PCBs.

3. To eliminate rough estimates with the exposure time, the design will have a constant time for the paper to be used like the acetate and tracing paper.

Significance of the Design

The features of this design will help the fabrication of PCBs become easier than the old one, eliminating the rough estimation of the exposure time of the PCBs through the use of preset time and buzzer. The exposure time of double sided PCBs will be minimized due to the capability of the device to support such kinds of PCBs. The design used Ultraviolet LEDs to support the optimal dimensions of the PCBs to be exposed, as well as to reduce energy consumption. These are some of the features that will surely benefit the electronic constructors.

The use of UV LED array in the innovation of the UV exposure box design will not only help lessen its cost, but it will also help the environment. Because LEDs are far more energy efficient than incandescent and fluorescent lamps, they also have greater life span than the other types of lamps. The aforementioned features do not only make the design a fully-featured box at a minimal cost; more importantly, this design can be considered an environment-friendly device.

Scope and Delimitations

The scope covers the lists of capabilities that the UV exposure box can perform while limitations are the operations that the device is restricted to execute.

The scope of the design includes the following:

1. Support of single sided PCBs.
2. Support of double sided PCBs.
3. Will expose only the optimal dimension of the PCB into the UV.
4. UV LEDs were used as the light source.
5. LCD monitor was used as user output interface.
6. Keypad was used as user input interface.
7. Constant time exposure for acetate.
8. Constant time exposure for tracing paper.
9. Buzzer was used to warn the user when the exposure time expires.

For the delimitations:

1. The device can only support up to 12” by 12” PCBs.
2. The device will not automatically detect the dimensions of the PCB.
3. The input for the PCB sizes is by inches and in whole number.
4. The presensitzed PCBs are the only kind of PCB to be used

5. The device cannot support irregular shaped PCB design

Definition of Terms

Alarm – It refers to a signal (as a loud noise or flashing light) that warns or alerts.

[Merriam Webster Dictionary]

Keypad – is a set of buttons arranged in a block which usually bear digits.

Light – is an electromagnetic radiation of a wavelength visible to human eye.

[Hoffman,2007]

Liquid Crystal Display (LCD**) –** It is a thin, flat display device made up of any number of colors or monochrome pixels arrayed in front of a light source or reflector.

[Alexander C. and Sadiku M. (2003). Fundamentals of Electric Ciruits, 2nd edition]

Presensitized PCB - A Presensitized PCB is a factory prepared PCB surface coated with a photo sensitive emulsion.

[PCB Design by Chris Stahl]

Printed Circuit Board (PCB**)** - used to mechanically support and electrically connect electronic components using conductive pathways, tracks or signal traces etched from copper sheets laminated onto a non-conductive substrate

[PCB Design by Chris Stahl]

Ultraviolet (UV**) -** is electromagnetic radiation with a wavelength shorter than that of visible light

[Charles S. 2007 / Electronics principles and applications, 7th edition. USA]

UV LED - A rectifying semiconductor device which converts electrical energy into electromagnetic radiation.

[Charles S. 2007 / Electronics principles and applications, 7th edition. USA]

CHAPTER 2

REVIEW OF RELATED DESIGN LITERATURES AND STUDIES

This chapter includes compiled research works and studies that are related to the design. The content of this chapter are used as reference for the development of the design. The group used the following related articles, research works, and inventions as reference which will aid in having an in-depth understanding about the design that the group is working on.

According to the article, "Presensitized PCB : Artwork transfer technique for a consistent and good results" the PCB is a vital part of the electronic circuit assembly, and the performance and reliability outcome of the circuit being built is dependent to a large extent how well the PCB was built. And to achieve the best result, the most practical and easiest way to do it is with the use of a presensitized PCB. A presensitized PCB is a factory prepared PCB surface coated with a photo sensitive emulsion. The presensitized PCB surface is protected by a thick gray film both from unwanted exposure and contamination. As long as the film is not removed, the PCB can take rough handling during cutting with minimal risk of damage. To finish the process, exposure box is needed to expose the PCB.

Since everyone in the group is an engineering student, everyone has experienced on how to fabricate PCBs, and this article helps the group to decide and pursue the design topic about exposure boxes.

Another basis of this design is the UV Exposure Box Light Project by Frank Cevic, dated July 2009. Which is a do it yourself project of UV Exposure Box Light because most commercial light boxes are very expensive. In this article, it is stated that use UVA lamps which emit safe wavelengths and are good for exposing PCBs. The tubes should either look like normal tubes or be dark blue/violet when unlit. And do not use UVC lamps which emit hazardous radiation and is no good for exposing PCBs. It is also stated all the materials used and the step by step procedure in building the box like the wood frames and how will the design look like.

This gives an idea to the group on what are the materials to be considered, and how to build the exposure box. This project also offers a thought for the group to make the existing design inexpensive.

The research conducted by Power Engineering Conference last 2009 joint by AUPEC 2009 Australasian Universities, which are the advantages of Light-

emitting diodes over Fluorescent tubes, gave the group a second thought on what materials to be used in the design. Because as stated on the research, fluorescent tubes have long been the mainstay of lighting for many premises, due to relatively high efficiency and long life. However, Light-emitting diode (LED) efficiencies are set to exceed those of fluorescent tubes, with dimming and controllable color rendering readily achievable.

Further research has been done by the group regarding with the lifetime of the light source to be used in the exposure box. As stated on the research, the lifetime of traditional light sources are rated through established test procedures. The life testing procedure for compact fluorescent lamps, for example, is published by the Illuminating Engineering Society (IES) as LM-65. It calls for a statistically valid sample of lamps to be tested at an ambient temperature of 25 degrees Celsius using an operating cycle of 3 hours ON and 20 minutes OFF. The point at which half the lamps in the sample have failed is the rated average life for that lamp. For 10,000 hour lamps, this process takes about 15 months.

While as for LED, life testing for Light-emitting diodes (LEDs) is impractical due to the long expected lifetime. Switching is not a determining factor in a Light-emitting diodes life, so there is no need for the on-off cycling used with other light sources. But even with 24/7 operation, testing an Light-emitting diode

for 50,000 hours would take 5.7 years. Because the technology continues to develop and evolve so quickly, products would be obsolete by the time they finished life testing.

However, a life testing procedure for Light-emitting diode is currently under development by the Illuminating Engineering Society of North America (IESNA). The proposed method is based on the idea of "useful life," i.e., the operating time in hours at which the device's light output has declined to a level deemed to no longer meet the needs of the application. For example, for general ambient lighting, the level might be set at 70% of initial lumens. Useful life would be stated as the average number of hours that the LED would operate before depreciating to 70% of initial lumens.

Light Source	Range of Typical Rated Life (hours)* (varies by specific lamp type)	Estimated Useful Life (L_{70})
Incandescent	750-2,000	
Halogen incandescent	3,000-4,000	
Compact fluorescent (CFL)	8,000-10,000	
Metal halide	7,500-20,000	
Linear fluorescent	20,000-30,000	
High-Power White LED		35,000-50,000

Table 2.1 Lifetime of Light Source

The research helps the group to use ultraviolet LEDs as alternative for ultraviolet lamps for the light source. LEDs are inexpensive and far more energy

efficient than lamps. In addition, the group decides to use the LEDs to control the light to be lit up by only exposing the optimal dimension of the PCB.

The paper entitled, "Packaging and powering of LED array" by Feng, Weifeng He, Yongzhi Shi in School of Engineering University of California Irvine mainly concerns about LED array. The paper stated some information that will benefit the group like the formation of the LED array, how LEDs are connected to each other, and what voltage to operate and how to manipulate the bit operations of the LED array.

The research about Scanning Led Array Based Volumetric Display by Saymta, M., Isikman, S.O., Urey, H., of Koc University Istanbul was about novel quasi-holographic display concept is developed using light emitting diode (LED) arrays on scanning platform. The display system is capable of providing smooth motion parallax and solving the accommodation-vergence rivalry. Each scanner module contains 1D LED array mounted on a polymer scanner with a lens for imaging the LEDs onto a special diffuser screen. The scanning modules are actuated electromagnetically and the LEDs are driven with an external LED driving field programmable gate array (FPGA) circuitry designed for the purpose. The scanners have a natural frequency of 12.7 Hz, scan line of 21.5deg total optical scan angle (TOSA) and a quality factor of 20.

The research will help the group on what method will the group use on how to manipulate the LEDs in exposing the optimal dimensions of the PCB. The scanning method is the best possible way on how to manipulate the LEDs.

One of the main objectives of the group is to reduce the energy consumption of the device. Based on the paper of Meralco Company, electricity in the Philippines is generated and passes through several entities. In this sense the cost of the price of electricity increases as it is generated, transferred, and distributed.

The following list are the breakdown of charges:

1. **Generation** (42.6 %)- This goes to the generating companies (National Power Corporation and Independent Power Producers) or power suppliers.
2. **Transmission** (9.4%) - This goes to the Transmission Company (TransCo).
3. **System Loss** (5.5%) - This is the recovery of the cost of power loss due to technical and non-technical system losses.
4. **Distribution** (31.5%) - These charges go to the Manila Electric Railroad and Light Company (Meralco) and cover the many services and tasks it performs, such as:
    a. The cost of building, operating and maintaining its distribution

system;

b. Billing, collection, customer service, records maintenance, and associated services;

c. Reading, operating and maintaining power metering facilities;

d. Adjustments for the unavoidable fluctuations in the exchange rate of the Philippine Peso against the US Dollar.

5. **Subsidies** (1.3%) - The subsidies under this group provide for socialized pricing mechanisms for marginalized customers and the different customer classifications.

6. **Goverment Taxes** (9.3%) and **Universal Charges** (0.4%) - These subgroups include taxes and other charges remitted to the national and local governments and the Power Sector Assets and Liabilities Management Corporation (PSALM). These remittances are used to fund the electrification of remote areas not connected to the transmission system.

As of the December 2010 – January 2011 billing of the Manila Electric Railroad and Light Company (Meralco):

1. Generation charge 4.7328 Pesos per Kilowatt-hour.

2. Transmission 1.0443 Pesos per Kilowatt-hour.

3. System loss 0.6110 Pesos per Kilowatt-hour.
4. Distribution cost 3.4996 Pesos per Kilowatt-hour.
5. Subsidies 0.1444 Pesos per Kilowatt-hour.
6. Government tax 1.0333 Pesos per Kilowatt-hour.
7. Universal Charge 0.0444 Pesos per Kilowatt-hour.

Giving a total of **11.1098 PHP** per Kilowatt-hour.

CHAPTER 3

DESIGN PROCEDURES

This type of design is a constructive research which means that it develops solution to a problem that has to be solved through the development of a system. The group needs to find ways on how to improve the existing ultraviolet exposure boxes. Improvement should include the reduction of cost, lessening of energy consumption, the support for double-sided PCB's, and the added features like the buzzer and the preset timer.

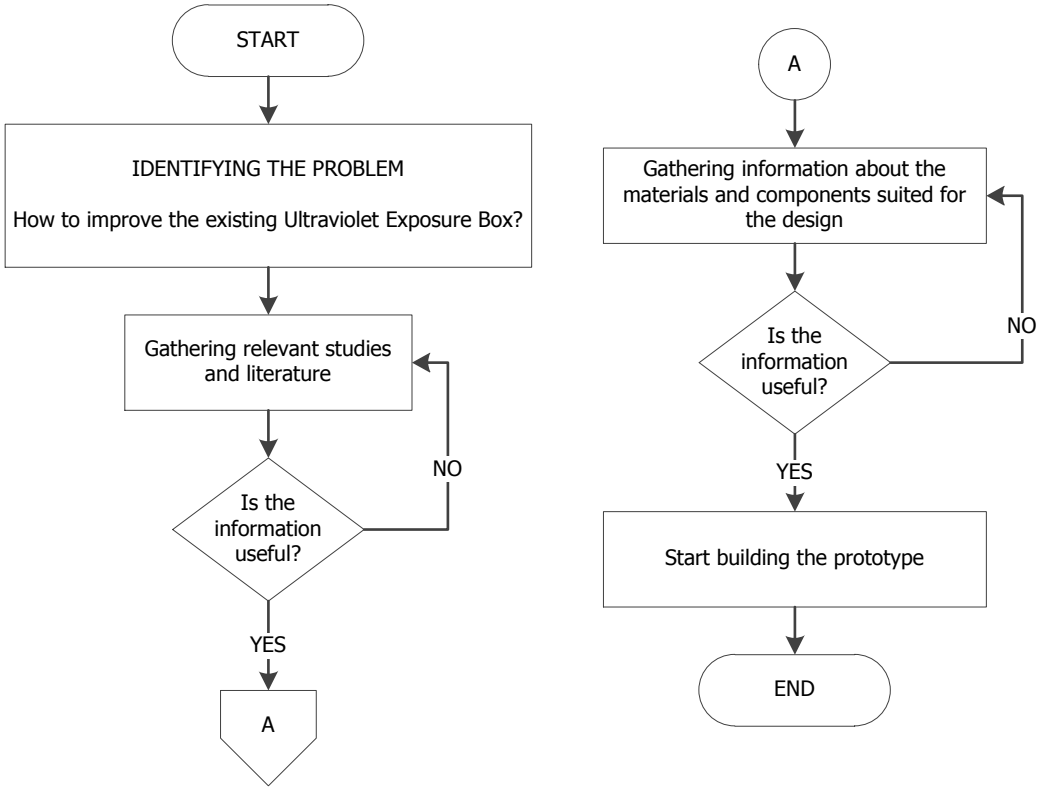


Figure 3.0 Design Methodology Flowchart

Figure 3.0 shows the procedure in developing the design. Identifying the problem would be the first step. In this case, brainstorming is done on how to come up with the design project. Questions on how the group can improve the existing ultraviolet exposure boxes were asked. Is it possible to replace the ultraviolet lamp with ultraviolet LEDs to lessen the cost and to minimize energy consumption by controlling the LEDs? Is it possible to add features like the support for double sided PCB's, buzzer for alarm, and preset time for duration of exposure to eliminate estimations?

The second step supports the first step. Gathering information related to the design would help to further develop the prototype. Previous designs were considered to have an idea on how to add an innovation to a new project.

Gathering information about the needed components was necessary. Adequate research of materials to back up the design was likewise done. Choosing the right microcontroller, LCD, keypad, buzzer, Ultraviolet LEDs, and others were accomplished. There were a few components that were bought but were not used during the actual creation. This was expected in the trial and error testing of the device.

Design Procedure for Actual Design

The prototype is made up of a controller box and an exposure box. All circuit components are in the controller box; from power supply, microcontroller, LCD display, keypad, up to the buzzer. The controller box is the user interface of the device which contains the keypad for selection in which the user will input all the information required. The exposure box, on the other hand, is where the ultraviolet LEDs and the glass will be placed. It is a box that could handle up to 12" by 12" PCB, in able to support double sided PCB the top and bottom part of the box are placed with the UV LED arrays. The UV LED arrays are controlled by the microcontroller which is placed inside the controller box.

Step-by-Step Design Procedure

1. Gather all necessary data and information needed for the design of the system.

2. Prepare all the materials needed for the assembly of the design.

3. Use circuit generating software to create a circuit design and PCB design.

4. Create a power supply with a bridge type rectifier connected to a voltage regulator. The rectifier will convert ac signal into dc signal. Connect it to a voltage regulator LM7805 that will produce another voltage 5V appropriate for some components like the microcontroller and LCD monitor.

5. Prepare the microcontroller PIC16F877 for programming and burning. The generated code is located at appendix C.

6. Connect all necessary components or other circuits to the microcontroller; power supply, keypads for selection keys, LCD display for visual interface, and the control for the LED arrays.

7. Prepare the materials needed for the exposure box; Ultraviolet LEDs, PCB, glass, etc.

8. Construct two 12” by 12” PCB for the UV LED array.

9. Use glass sheet rather than plastic for the top of the UV unit, as it will flex less and be less prone to scratches.

10. Place the UV LED array 3” away from the glass (top and bottom part).

11. Connect all other necessary components.

12. Test all parts of the design. Conduct troubleshooting to verify the output of the design.

13. Construct the final casing for the design.

Hardware Development

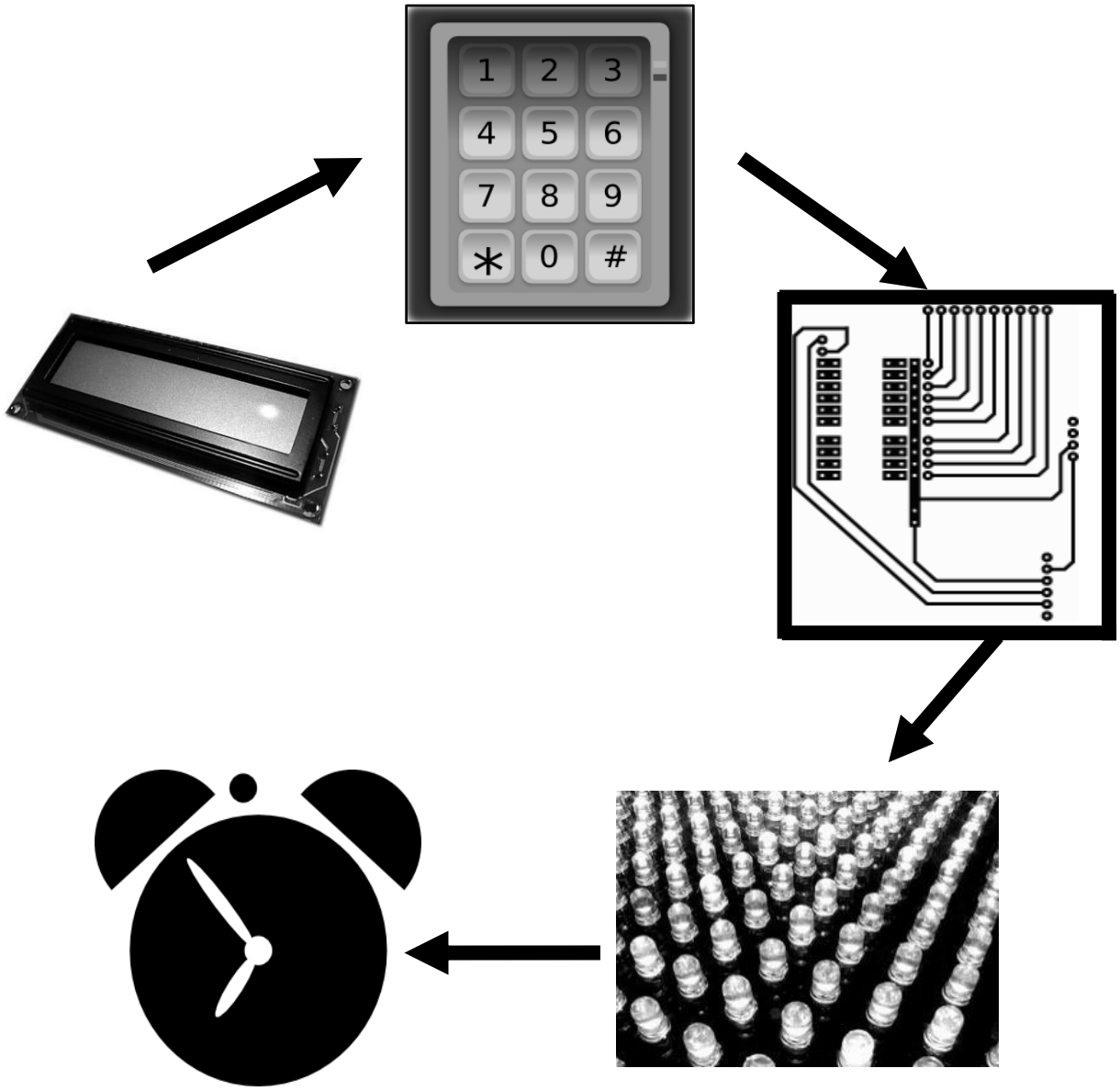


Figure 3.1 Conceptual Diagram

Figure 3.1 shows the basic concept of the design. The LCD screen displays all required information that is first needed to be filled-up with the keypad. After providing the required information, the presensitized PCB together with the layout will be exposed to the UV LED array. Finally a buzzer will be triggered when the preset exposure time is over.

Block Diagram

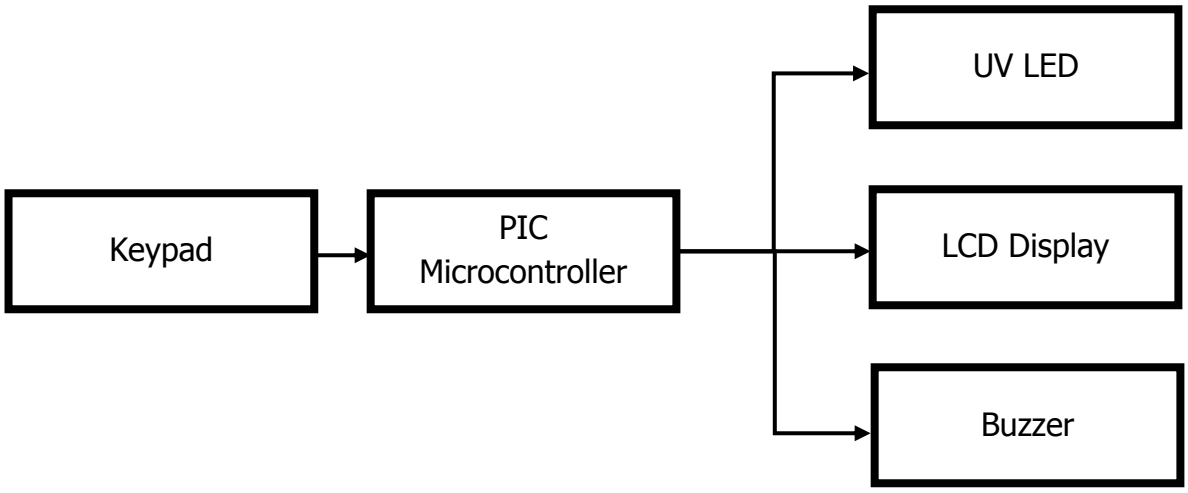


Figure 3.2 Block Diagram

Inputs are done with the use of a keypad, all information that were entered will be passed to the microcontroller. The PIC Microcontroller is the one responsible for showing the settings in the LCD display, for triggering the UV LED array to lit up and expose the PCB, also for activating the buzzer when the exposure time expires. It is like the brain of the whole circuit for it processes all the controls inside the design.

USER INPUT

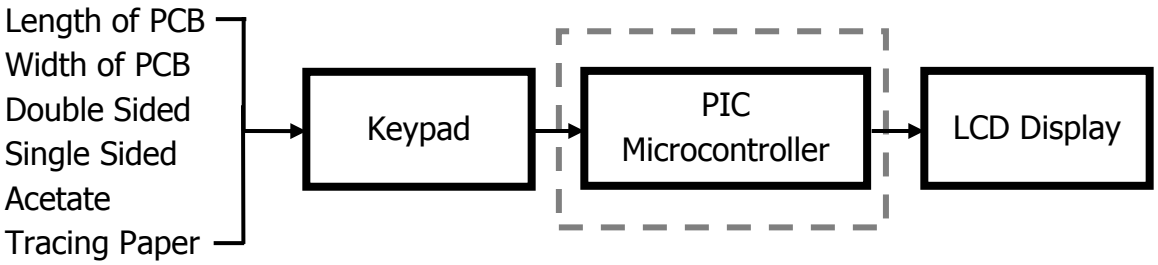


Figure 3.2.1 User Input

TYPE OF PCB

a. Single Sided

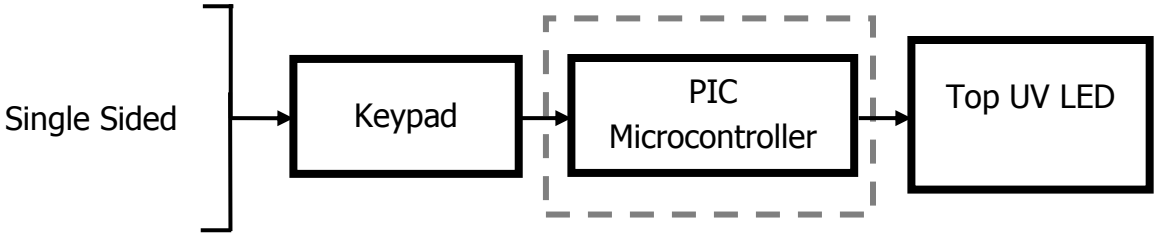


Figure 3.2.2 Single-sided

If the user selects single sided as the type of the PCB, the microprocessor will process the information and will let the bottom of the UV Box to lit up and expose the PCB.

b. Double Sided

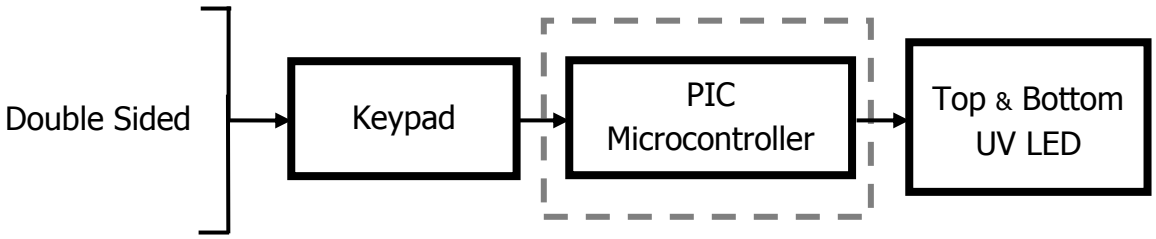


Figure 3.2.3 Double-sided

If the user selects double sided as the type of the PCB, the microprocessor will process the information and will let the top and the bottom of the UV Box to lit up and expose the PCB.

TYPE OF PAPER

a. Acetate

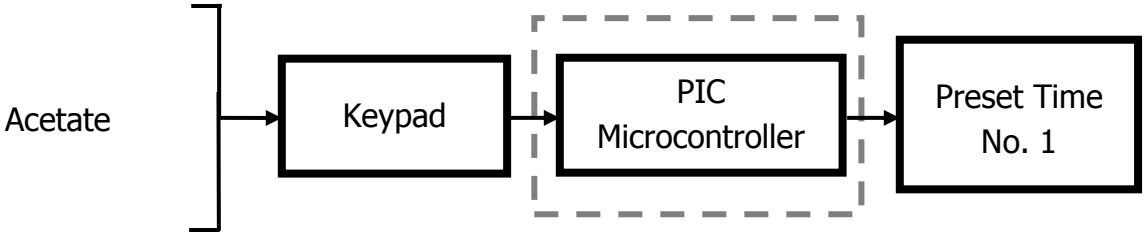


Figure 3.2.4 Acetate

If the user selects acetate as the type of the material to be used in exposure of the PCB, the microprocessor will process this information and then will activate a preset time that will allow the PCB to be exposed in a limited time.

b. Tracing Paper

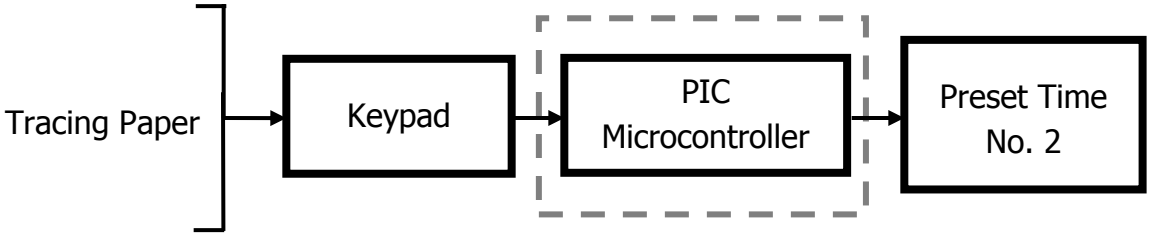


Figure 3.2.5 Tracing Paper

If the user selects tracing paper as the type of the material to be used in exposure of the PCB, the microprocessor will process this information and then will activate a preset time that will allow the PCB to be exposed in a limited time.

SCHEMATIC DIAGRAM

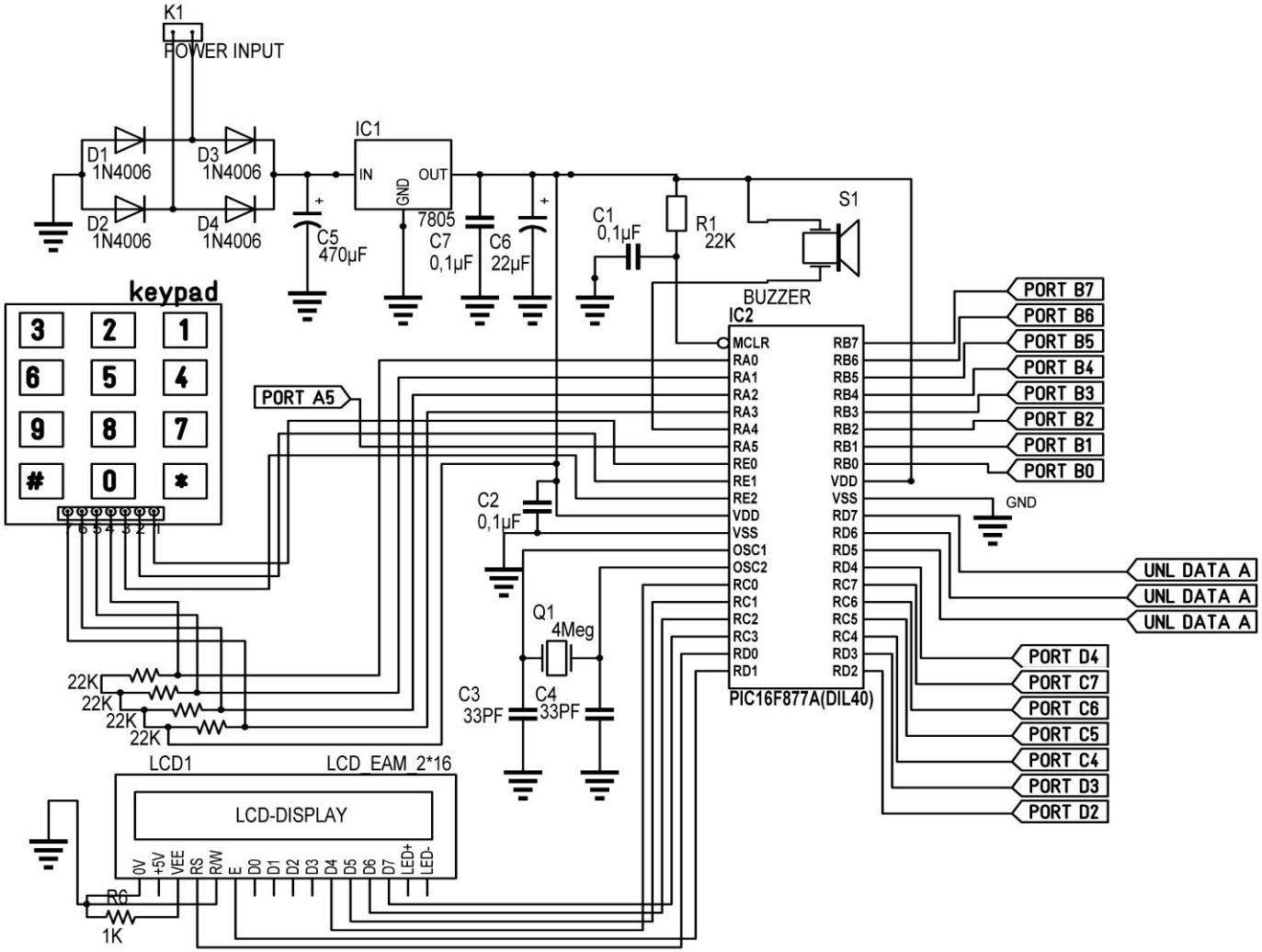


Figure 3.3 Schematic Diagram

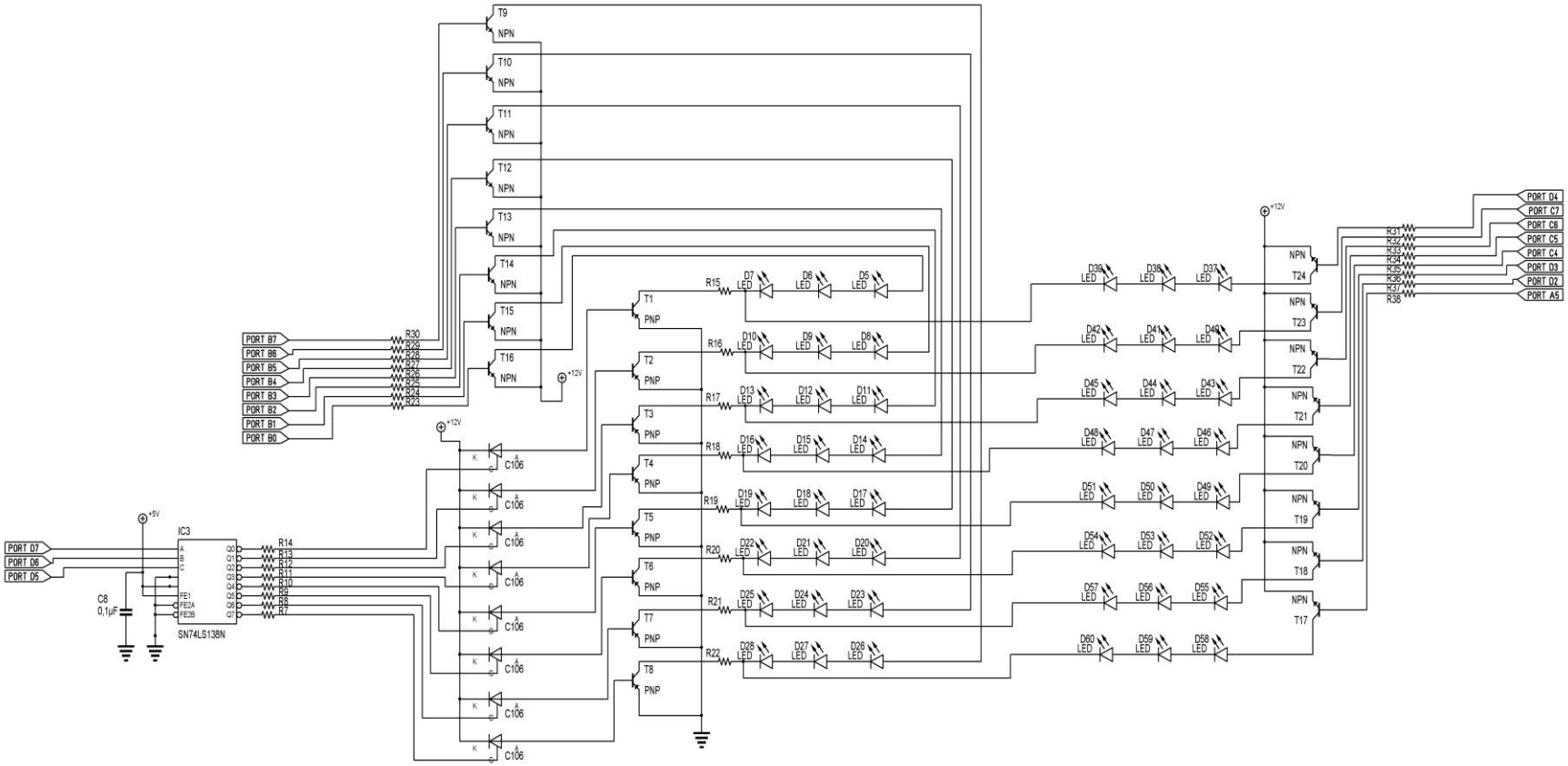


Figure 3.3.1 Schematic Diagram

Figure 3.3.0 and figure 3.3.1 shows the whole schematic diagram of the UV LED Exposure Box. A 2x16 LCD display and a keypad is directly connected to the microcontroller, their difference is that LCD display serves as the main interface of the prototype, while the keypad act as the input for the microcontroller. The microcontroller will determine and interpret every input that will be entered on the keypad. For pin 13 and 14, a crystal oscillator is connected having two capacitor values of 33pF to form the load capacitor for the crystal. Vpp pin or pin is connected to +5V, having a 22KΩ resistor to limit the current, which will power up the microcontroller. After the microcontroller, the transistors are placed to amplify the +5V supply coming from the microcontroller into +12V to have the best possible intensity of light for each ultraviolet LEDs. 74LS138 or the multiplexer IC was used also for the control of the LED array, it also helps to utilize the number of pins of the microcontroller efficiently. Flat top type ultraviolet LEDs are used in the prototype to have better exposure coverage because it has a wider viewing angle than circular shaped LEDs.

Software Development

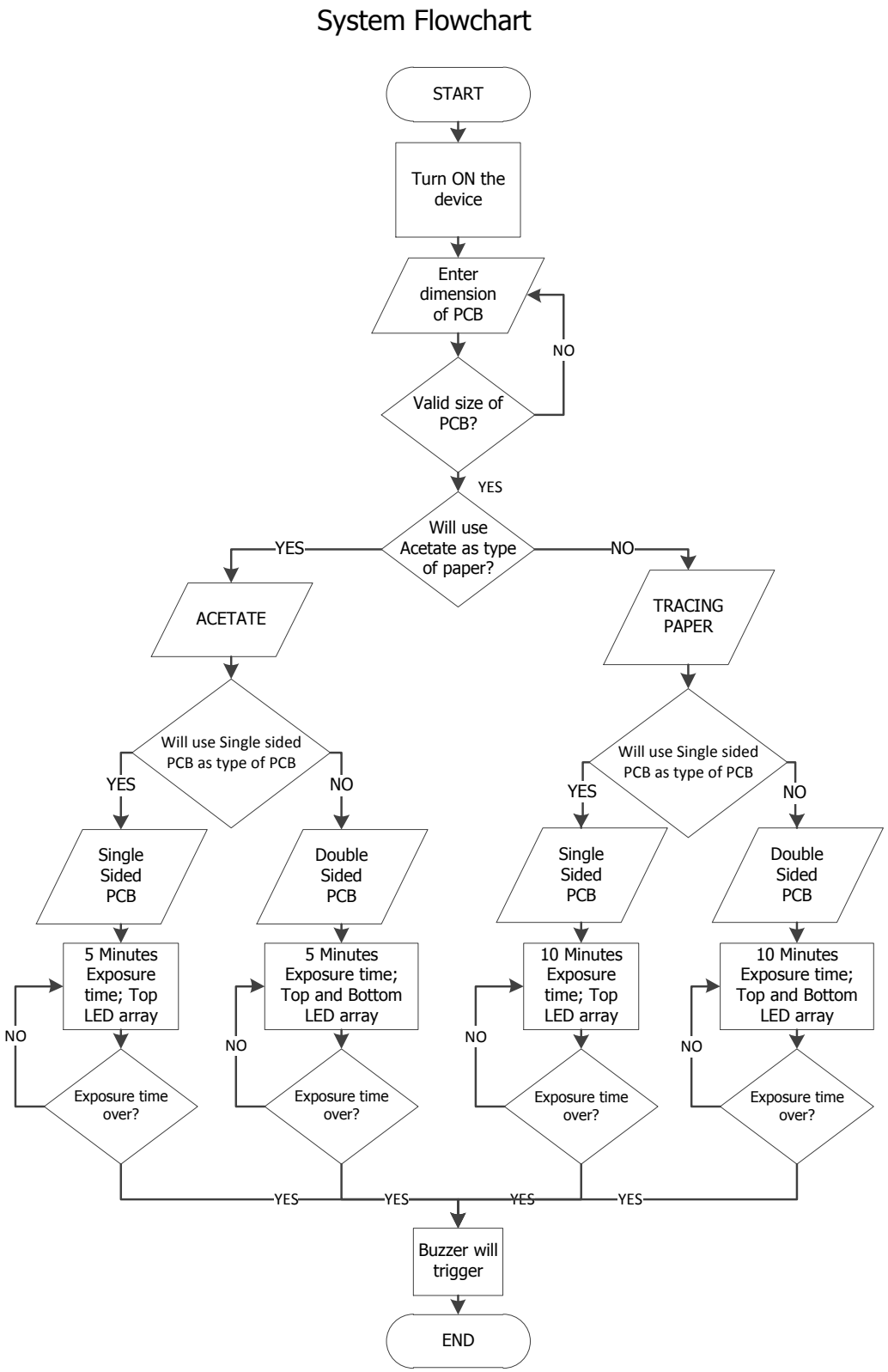


Figure 3.4 System Flowchart

Prototype Development

The ultraviolet LED exposure box consists of two major components, the controller box and the exposure box. All circuit components are in the controller box; from power supply, microcontroller, keypad, LCD module, transistors up to the buzzer. The exposure box, on the other hand, is where the ultraviolet LEDs and the glass will be placed.

Decomposition of Components

Parts name	Quantity	Price in Peso	Total
PIC 16F877A	1	320.00	320.00
IC Socket – 40 pins	1	15.00	15.00
4Mhz Crystal Oscillator	1	30.00	30.00
LCD 2 x 16 Display	1	450.00	450.00
Keypad	1	165.00	165.00
Buzzer	1	30.00	30.00
74LS138	1	30.00	30.00
IC Socket – 16 pins	2	8.00	16.00
Ultra-Violet LED	226	4.00	904.00
C4149	16	30.00	480.00
TIP32	8	30.00	240.00
PCB	2	180.00	360.00
Transformer 1A	1	200.00	200.00
7805	1	30.00	30.00
Rectifier Diode	1	30.00	30.00
2 Pins Connector Male/Female	2	10.00	20.00

C106	8	30.00	240.00
3 Pins Connector Male/Female	2	12.00	24.00
8 Pins Connector Male/Female	7	25.00	175.00
1/2W – 100 Ohm Resistor	10	2.00	20.00
1/4W – 1K Ohm Resistor	32	1.00	32.00
1/4W – 22K Ohm Resistor	4	1.00	4.00
5 meter wires	1	25.00	25.00
33pF Capacitor	1	2.00	2.00
16V-470 Capacitor	1	10.00	10.00
35V-22 Capacitor	1	5.00	5.00
35V-2200	1	30	30.00
Casing	1	300.00	300.00
		Total:	4,187.00

Table 3.1 Decomposition of Components

PIC16F877 Microcontroller

PIC16F877 is a microcontroller with 40 I/O pins which is suitable for the design because the design requires a lot of ports to support all the function of the prototype.

Keypad

Keypad is responsible for the entering all required information, such as the size of the PCB, paper to be used, and the type of the PCB.

Asterisk (*) – Asterisk is being used for the confirmation of process.

Sharp (#) – Sharp is being used for cancelation of process.

LCD Module

The LCD module is the one responsible for displaying the required information that are needed to be filled up.

The LCD module in idle state shows:

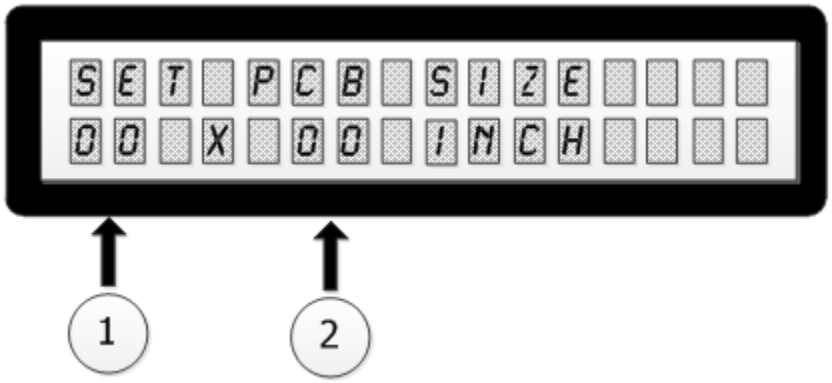


Figure 3.5.1 LCD Idle State

1. Height of the PCB
2. Width of the PCB

After entering the desired dimension of the PCB the confirmation screen appears

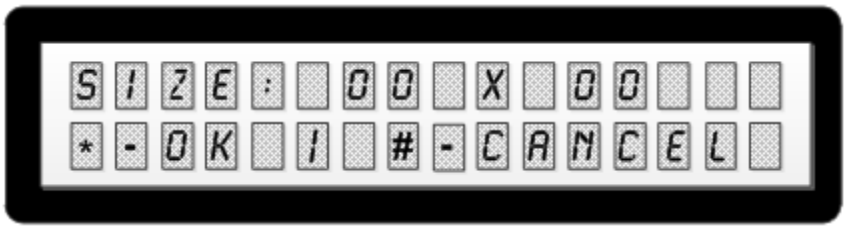


Figure 3.5.2 LCD Confirmation Screen

If the entered size of PCB is beyond 12’ by 12’ PCB it will display invalid size.

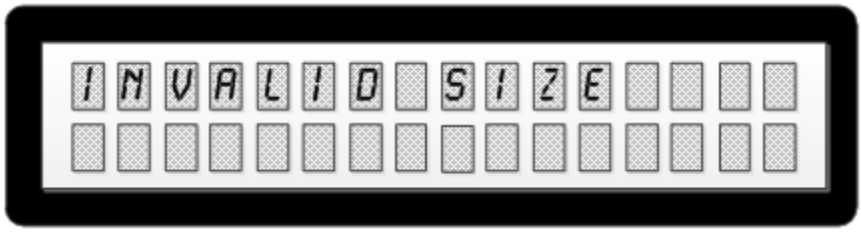


Figure 3.5.3 Invalid Size

The next step is to choose the type of paper so the LCD will display acetate or tracing paper.

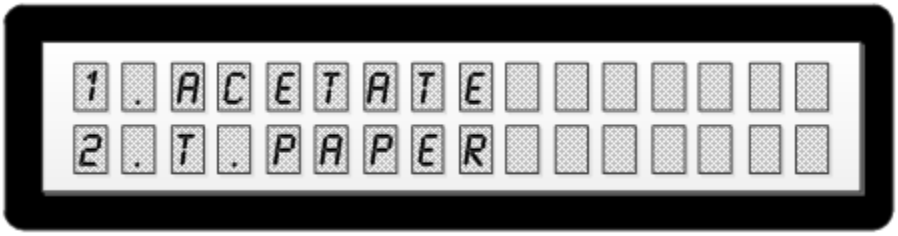


Figure 3.5.4 Type of Paper

The last step would be choosing for the type of PCB to be used, so the PCB will display single-sided or double-sided PCB.

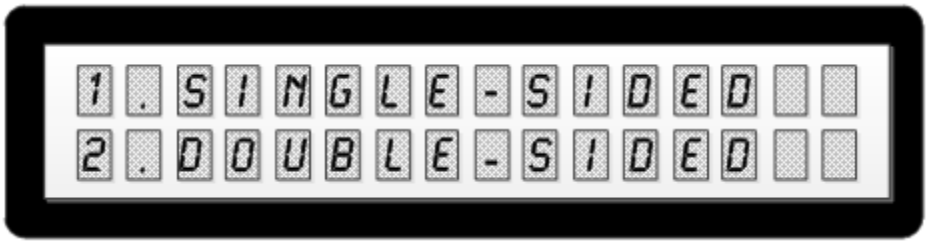


Figure 3.5.5 Type of PCB

After entering all necessary information, a timer will appear on the LCD display.

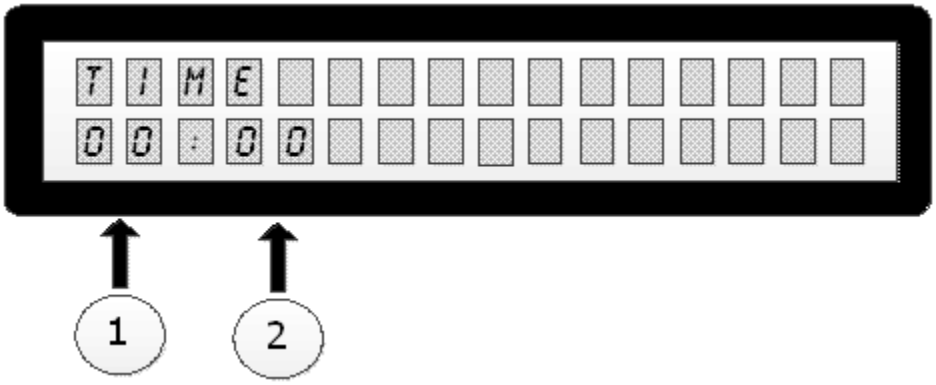


Figure 3.5.6 Timer

1. Minutes – maximum exposure time in minutes for acetate is 5

   - maximum exposure time in minutes for tracing paper is 10

2. Seconds

Buzzer

The buzzer will alarm the user when the process of PCB exposure is over.

CHAPTER 4

TESTING, PRESENTATION, AND INTERPRETATION OF DATA

This chapter gives the details on how the system was tested in relation to the design objectives.

Ultraviolet LEDs as Light Source

The UV LED exposure box can support PCB sizes up to 12" by 12" PCB. The Ultraviolet LEDs capability as the light source was tested by using different sizes of PCB's to be exposed. Random sizes ranging from 1" by 1" PCB up to 12" by 12" PCB were taken for sample. The support for the single-sided and double-sided PCB was some of the additional features of the prototype. This test will also check the capability of the prototype in supporting double sided PCB's.

Table 4.1 shows the parameters to be considered in the test (e.g. sizes of the PCB and the type of PCB). It also shows the results of the conducted test whether the PCB was successfully exposed or not.

TYPE OF PCB	SIZE OF THE PCB	Is the PCB was successfully exposed?
Single/Double	0	Yes or No
Single-sided	1" by 6"	Yes
Single-sided	6" by 7"	Yes
Single-sided	4" by 3"	Yes
Single-sided	4" by 6"	Yes

Double-sided	1″ by 1″	Yes
Double-sided	2″ by 1″	Yes
Double-sided	2″ by 2″	Yes
Double-sided	2″ by 8″	Yes
Double-sided	2″ by 5″	Yes
Double-sided	12″ by 12″	Yes

Table 4.1 UV LEDs as Light Source

Table 4.1 shows the testing of different sizes of PCB's exposed to Ultraviolet LEDs. During the test, the prototype was observed whether the Ultraviolet LEDs can expose the optimal dimension of the PCB's. Based on the gathered data, the capability of the Ultraviolet LED to expose the PCB worked properly. Also, the support for double sided PCB's worked as it should be.

Ultraviolet LEDs for Less Energy Consumption

After some research, the group found out that LEDs are far more energy efficient than either incandescent or fluorescent lamps, making them cheaper to run and kinder to the environment. Also according to some articles, LEDs have a far greater life span than the other types of lamp measured in decades rather than months. So the group decided to use UV LEDs rather than UV lamps to lessen the cost and reduce energy consumption.

This test will show the comparison of data gathered between the original UV PCB exposure boxes and the design of the group. The parameters to be considered in this test are the following:

1. Type of PCB to be exposed
2. Length of Exposure time
3. Price in peso per kilowatt hour

Type of PCB	Voltage Used	Current Reading	Power (Watts)	Exposure Time	PHP/kwh	**Total Cost/Use**
Double Sided	220V	0.73A	160.6W	100sec	11.1099	0.1784PHP
Single Sided	220V	0.38A	83.6W	100sec	11.1099	0.0929PHP

Table 4.2 UV Lamp PCB Exposure Box

Type of PCB	Voltage Used	Current Reading	Power (Watts)	Exposure Time	PHP/kwh	**Total Cost/Use**
Double Sided	220V	0.032A	7.04W	300sec	11.1099	0.0065PHP
Single Sided	220V	0.029A	6.38W	300sec	11.1099	0.0059PHP

Table 4.3 UV LED PCB Exposure Box

To obtain the results as shown in Table 4.2 and Table 4.3, formulas that were used are the following:

$$P = I \times V$$

$$Number\ of\ Hours = \left(\frac{1\ hour}{3600 sec}\right) \times (Exposure\ time)$$

$$Total\ cost\ per\ use = \left(\frac{P}{1000}\right) \times \left(\frac{PHP}{kwh}\right) \times (Number\ of\ Hours)$$

Table 4.2 shows the performance of the commercialize UV exposure box that uses ultraviolet lamp as the light source. While table 4.3 shows the performance of the design that the group has developed which uses ultraviolet LEDs as alternative for UV lamps for the light source.

Based from the gathered data after performing the test, it shows that using ultraviolet LEDs as the light source is far more efficient than using ultraviolet lamp. LEDs were efficient in terms of cost and power consumption. In addition, the use of the ultraviolet LEDs helps the design to control the lights to be lit up to expose the optimal dimension of the PCB.

Preset Exposure Time

Exposure time in ultraviolet light is one of the major requirements in fabricating presensitized PCB. The group conducted series of test concerning with the exposure time to obtain the best results of fabricated PCB.

Table 4.4 and table 4.5 shows the trial and error experiment in acquiring the best exposure time to obtain the top results in fabrication of presensitized PCB. This test aims to eliminate the rough estimation of exposure time to reduce the errors in fabricating presensitized PCB. There were few PCB's that were bought and wasted during the experiment. This was expected in the trial and error testing of the device.

Exposure Time	RESULT
Time (sec)	Under-exposed/ Normal/ Over-exposed
100 sec	Under-exposed
600 sec	Over-exposed
300 sec	Normal

Table 4.4 Preset time for Acetate

Exposure Time	RESULT
Time (sec)	Under-exposed/ Normal/ Over-exposed
300 sec	Under-exposed
600 sec	Normal

Table 4.4 Preset time for Tracing Paper

UNDER-EXPOSED – layout is not visible / lines are not clear; too short exposure time

OVER-EXPOSED – too long exposure time; turns the PCB to black

NORMAL – layouts and lines are visible

Separate tests were conducted for the exposure time for acetate and tracing paper.

Based on the gathered data on the test for acetate, under or over 300 seconds of exposure time doesn't give the best result of fabricated PCB. To get the best possible fabricated PCB, the group set the constant exposure time for the acetate into 300 seconds.

While for the tracing paper, 300 seconds is not enough to expose the PCB because of the thickness of the tracing paper. The group doubled the exposure time for the tracing paper to determine whether it will give the best result. 600 seconds of exposure time gave the best result in exposing the PCB using the

tracing paper. So the group set the constant exposure time for tracing paper to 600 seconds.

Having the constant exposure time for each type of paper, the electronic constructor will no longer have to guess what will be the best duration of exposure time.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

CONCLUSION

The group found out that the design is capable of providing electronic constructors an effective and low cost power-saving UV exposure box. Like the existing UV exposure boxes, the design was successful in exposing the PCB into the ultraviolet light for it to be developed. The alternative of using ultraviolet LEDs instead of ultraviolet lamps helps the design to lessen the overall cost and to reduce energy consumption. Moreover exposure for the optimal sizes of PCB was achieved by controlling the ultraviolet LEDs. Furthermore, the regular process of exposing the presensitized PCB had been made easier when the design was added with the support for double-sided PCBs and preset time for exposure depending on the type of paper.

RECOMMENDATION

Improvements on the design can be applied to further enhance its capabilities and functionalities, like having a sensor that will automatically detect the size of the PCB. Having the sensor, the design will no longer require the user to input the dimension of the PCB to be exposed. Furthermore, having the

sensor installed into the prototype, the design can support other PCB design that comes with irregular shaped layouts. Another improvement that can be done for the project is its weight and bulkiness. Finding a way to have a case that is more compact can be the solution for these problems.

BIBLIOGRAPHY

Floyd, Thomas L., 1991. Electronics Fundamentals: Circuits, Devices and Applications, 2nd Edition, New York: Macmillan Publishing Co.

Saymta, M. Isikman, S.O. Urey, H. Koc Univ., Istanbu, Scanning Led Array Based Volumetric Display, 28-30 May 2008 , On page(s): 21

Hilbourne, R.A. Jones, D.D. , Transistor power amplifiers , Proceedings of the IEE - Part B: Radio and Electronic Engineering Issue Date : November 1955 Volume : 102 , Issue:6

Bergmann, M. Kuhr, T. Haberem, K. Hussell, C. Abare, A. Emerson, D. Cree, Inc, Durham, NC, USA , Materials and device developments for ultraviolet LEDs and laser diodes, Device Research Conference, 2004. 62nd DRC. Conference Digest [Includes 'Late News Papers' volume] Issue Date : 21-23 June 2004 On page(s): 151

Lim Siong Boon, PCB Board Fabrication, DIY and fabricate your own PCB board at home, Dated: 08-Dec-09

APPENDIX A

Operation's Manual

User's Manual

a) Plug the power supply of the device into the 220v outlet.

b) Place the PCB and the layout in the lowest and rightmost part of the glass. Face the PCB upward when it is a single sided PCB.

c) Set the dimension of the PCB using the keypad. Every digit must be followed up by an asterisk (*) for confirmation.

d) A confirmation screen for the size of the PCB will be displayed. Press asterisk (*) to confirm, on the other hand press sharp (#) to cancel the process.

e) Choose what type of paper to be used. Press #1 for acetate or press #2 for tracing paper.

f) Next, choose what type of PCB to be used. Press #1 for single-sided PCB or press #2 for double-sided PCB.

g) Wait for the buzzer to trigger, it means the exposure process is over.

h) When done using the device, remove the plug from the outlet.

APPENDIX B

PICTURE OF PROTOTYPE

Figure 6.1 Prototype

Figure 6.2 Controller Box

Figure 6.3 LED Panels (TURNED OFF)

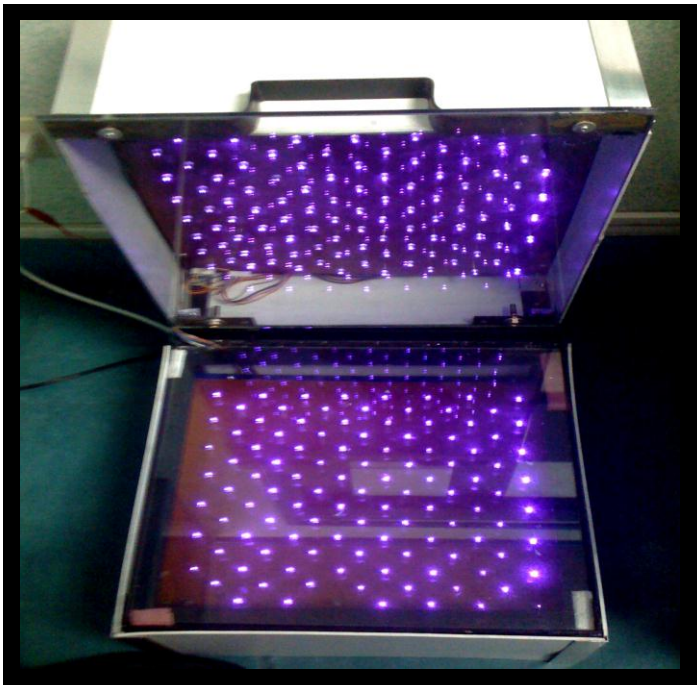


Figure 6.4 LED Panels (TURNED ON)

APPENDIX C

PROGRAM LISTING

```
;-----------------------------------------------------------
; Code Produced by the Proton+ 3.4.0.8
; Copyright Rosetta Technologies/Crownhill Associates Ltd
; Written by Les Johnson. October 2009
;-----------------------------------------------------------
;
 LIST  P = 16F877A, F = INHX8M, W = 2, X = ON, R = DEC, MM = ON, N = 0,
C = 255
INDF EQU 0X0000
TMR0 EQU 0X0001
PCL EQU 0X0002
STATUS EQU 0X0003
FSR EQU 0X0004
PORTA EQU 0X0005
PORTB EQU 0X0006
PORTC EQU 0X0007
PORTD EQU 0X0008
PORTE EQU 0X0009
PCLATH EQU 0X000A
INTCON EQU 0X000B
PIR1 EQU 0X000C
PIR2 EQU 0X000D
TMR1L EQU 0X000E
TMR1LH EQU 0X000F
TMR1H EQU 0X000F
T1CON EQU 0X0010
TMR2 EQU 0X0011
T2CON EQU 0X0012
SSPBUF EQU 0X0013
SSPCON EQU 0X0014
CCPR1L EQU 0X0015
CCPR1LH EQU 0X0016
CCPR1H EQU 0X0016
CCP1CON EQU 0X0017
RCSTA EQU 0X0018
TXREG EQU 0X0019
RCREG EQU 0X001A
CCPR2L EQU 0X001B
CCPR2LH EQU 0X001C
CCPR2H EQU 0X001C
CCP2CON EQU 0X001D
ADRESLH EQU 0X001E
ADRESH EQU 0X001E
ADCON0 EQU 0X001F
OPTION_REG EQU 0X0081
TRISA EQU 0X0085
TRISB EQU 0X0086
TRISC EQU 0X0087
TRISD EQU 0X0088
TRISE EQU 0X008
```

```
PIE1 EQU 0X008C
PIE2 EQU 0X008D
PCON EQU 0X008E
SSPCON2 EQU 0X0091
PR2 EQU 0X0092
SSPADD EQU 0X0093
SSPSTAT EQU 0X0094
TXSTA EQU 0X0098
SPBRG EQU 0X0099
CMCON EQU 0X009C
CVRCON EQU 0X009D
ADRESL EQU 0X009E
ADCON1 EQU 0X009F
EEDATL EQU 0X010C
EEDATA EQU 0X010C
EEDAT EQU 0X010C
EEADR EQU 0X010D
EEDATLH EQU 0X010E
EEDATH EQU 0X010E
EEADRH EQU 0X010F
EECON1 EQU 0X018C
EECON2 EQU 0X018D
_I2C_SCL_PORT = TRISC
_I2C_SCL_PIN = 3
_I2C_SDA_PORT = TRISC
_I2C_SDA_PIN = 4
IRP=7
RP1=6
RP0=5
NOT_TO=4
NOT_PD=3
Z=2
DC=1
C=0
GIE=7
PEIE=6
T0IE=5
TMR0IE=5
INTE=4
RBIE=3
T0IF=2
TMR0IF=2
INTF=1
RBIF=0
PSPIF=7
ADIF=6
RCIF=5
PP_RCIF=5
TXIF=4
PP_TXIF=4
SSPIF=3
CCP1IF=2
TMR2IF=1
TMR1IF=0
CMIF=6
EEIF=4
BCLIF=3
CCP2IF=0
T1CKPS1=5
T1CKPS0=4
T1OSCEN=3
NOT_T1SYNC=2
T1INSYNC=2
T1SYNC=2
TMR1CS=1
TMR1ON=0
TOUTPS3=6
TOUTPS2=5
TOUTPS1=4
TOUTPS0=3
TMR2ON=2
PP_TMR2ON=2
T2CKPS1=1
PP_T2CKPS1=1
T2CKPS0=0
PP_T2CKPS0=0
WCOL=7
SSPOV=6
SSPEN=5
CKP=4
SSPM3=3
SSPM2=2
SSPM1=1
SSPM0=0
CCP1X=5
CCP1Y=4
CCP1M3=3
CCP1M2=2
CCP1M1=1
CCP1M0=0
SPEN=7
RX9=6
RC9=6
NOT_RC8=6
RC8_9=6
SREN=5
CREN=4
PP_CREN=4
ADDEN=3
FERR=2
OERR=1
PP_OERR=1
RX9D=0
RCD8=0
CCP2X=5
CCP2Y=4
CCP2M3=3
CCP2M2=2
CCP2M1=1
CCP2M0=0
ADCS1=7
ADCS0=6
CHS2=5
CHS1=4
```

```
CHS0=3
GO=2
NOT_DONE=2
GO_DONE=2
PP_GO_DONE=2
ADON=0
NOT_RBPU=7
INTEDG=6
T0CS=5
T0SE=4
PSA=3
PS2=2
PS1=1
PS0=0
IBF=7
OBF=6
IBOV=5
PSPMODE=4
TRISE2=2
TRISE1=1
TRISE0=0
PSPIE=7
ADIE=6
RCIE=5
TXIE=4
SSPIE=3
CCP1IE=2
TMR2IE=1
TMR1IE=0
CMIE=6
EEIE=4
BCLIE=3
CCP2IE=0
NOT_POR=1
NOT_BO=0
NOT_BOR=0
GCEN=7
ACKSTAT=6
PP_ACKSTAT=6
ACKDT=5
ACKEN=4
PP_ACKEN=4
RCEN=3
PP_RCEN=3
PEN=2
PP_PEN=2
RSEN=1
PP_RSEN=1
SEN=0
PP_SEN=0
SMP=7
CKE=6
D=5
I2C_DATA=5
NOT_A=5
NOT_ADDRESS=5
D_A=5
DATA_ADDRESS=5
P=4
I2C_STOP=4
S=3
I2C_START=3
R=2
I2C_READ=2
NOT_W=2
NOT_WRITE=2
R_W=2
PP_R_W=2
READ_WRITE=2
UA=1
BF=0
PP_BF=0
CSRC=7
TX9=6
NOT_TX8=6
TX8_9=6
TXEN=5
SYNC=4
BRGH=2
TRMT=1
TX9D=0
TXD8=0
C2OUT=7
C1OUT=6
C2INV=5
C1INV=4
CIS=3
CM2=2
CM1=1
CM0=0
CVREN=7
CVROE=6
CVRR=5
CVR3=3
CVR2=2
CVR1=1
CVR0=0
ADFM=7
ADCS2=6
PCFG3=3
PCFG2=2
PCFG1=1
PCFG0=0
EEPGD=7
PP_EEPGD=7
WRERR=3
WREN=2
PP_WREN=2
WR=1
PP_WR=1
RD=0
PP_RD=0
  __MAXRAM 0X1FF
```

```
  __BADRAM 0X8F-0X90, 0X95-0X97,
0X9A-0X9B
  __BADRAM 0X105, 0X107-0X109
  __BADRAM 0X185, 0X187-0X189,
0X18E-0X18F
CP_ALL EQU 0X1FFF
CP_OFF EQU 0X3FFF
DEBUG_OFF EQU 0X3FFF
DEBUG_ON EQU 0X37FF
WRT_OFF EQU 0X3FFF
WRT_256 EQU 0X3DFF
WRT_1FOURTH EQU 0X3BFF
WRT_HALF EQU 0X39FF
CPD_OFF EQU 0X3FFF
CPD_ON EQU 0X3EFF
LVP_ON EQU 0X3FFF
LVP_OFF EQU 0X3F7F
BODEN_ON EQU 0X3FFF
BODEN_OFF EQU 0X3FBF
PWRTE_OFF EQU 0X3FFF
PWRTE_ON EQU 0X3FF7
WDT_ON EQU 0X3FFF
WDT_OFF EQU 0X3FFB
RC_OSC EQU 0X3FFF
HS_OSC EQU 0X3FFE
XT_OSC EQU 0X3FFD
LP_OSC EQU 0X3FFC
#DEFINE __16F877A 1
#DEFINE XTAL 4
#DEFINE _CORE 14
#DEFINE _MAXRAM 368
#DEFINE _RAM_END 368
#DEFINE _MAXMEM 8192
#DEFINE _ADC 8
#DEFINE _ADC_RES 10
#DEFINE _EEPROM 256
#DEFINE _PAGES 4
#DEFINE _BANKS 3
#DEFINE RAM_BANKS 4
#DEFINE _USART 1
#DEFINE _USB 0
#DEFINE _FLASH 1
#DEFINE _CWRITE_BLOCK 8
#DEFINE BANK0_START 32
#DEFINE BANK0_END 123
#DEFINE BANK1_START 160
#DEFINE BANK1_END 239
#DEFINE BANK2_START 272
#DEFINE BANK2_END 367
#DEFINE BANK3_START 400
#DEFINE BANK3_END 495
#DEFINE _SYSTEM_VARIABLE_COUNT 30
RAM_BANK = 0
CURRENT@PAGE = 0
DEST@PAGE = 0
#DEFINE LCD#DTPORT PORTC
#DEFINE LCD#DTPIN 0
#DEFINE LCD#RSPORT PORTD
#DEFINE LCD#RSPIN 0
#DEFINE LCD#ENPORT PORTD
#DEFINE LCD#ENPIN 1
#DEFINE LCD#INTERFACE 4
#DEFINE LCD#LINES 2
#DEFINE LCD#TYPE 0
F@CALL MACRO DEST
IF (DEST < 1)
IF ((DEST & 2048) == 0)
        BCF 10,3
ELSE
        BSF 10,3
ENDIF
IF ((DEST & 4096) == 0)
        BCF 10,4
ELSE
        BSF 10,4
ENDIF
ELSE
IF (DEST > $)
IF ((DEST & 2048) == 0)
        BCF 10,3
ELSE
        BSF 10,3
ENDIF
IF ((DEST & 4096) == 0)
        BCF 10,4
ELSE
        BSF 10,4
ENDIF
ELSE
IF ((DEST & 6144) == 0)
        CLRF 10
ELSE
IF ((DEST & 2048) == 0)
        BCF 10,3
ELSE
        BSF 10,3
ENDIF
IF ((DEST & 4096) == 0)
        BCF 10,4
ELSE
        BSF 10,4
ENDIF
ENDIF
ENDIF
ENDIF
        CALL DEST
        ENDM
F@JUMP MACRO DEST
IF (DEST < 1)
IF ((DEST & 2048) == 0)
        BCF 10,3
ELSE
        BSF 10,3
ENDIF
```

```
IF ((DEST & 4096) == 0)
        BCF 10,4
ELSE
        BSF 10,4
ENDIF
ELSE
IF (DEST > $)
IF ((DEST & 2048) == 0)
        BCF 10,3
ELSE
        BSF 10,3
ENDIF
IF ((DEST & 4096) == 0)
        BCF 10,4
ELSE
        BSF 10,4
ENDIF
ELSE
IF ((DEST & 6144) == 0)
        CLRF 10
ELSE
IF ((DEST & 2048) == 0)
        BCF 10,3
ELSE
        BSF 10,3
ENDIF
IF ((DEST & 4096) == 0)
        BCF 10,4
ELSE
        BSF 10,4
ENDIF
ENDIF
ENDIF
ENDIF
        GOTO DEST
        ENDM
SET@PAGE MACRO DEST
IF ((DEST & 2048) == 0)
        BCF 10,3
ELSE
        BSF 10,3
ENDIF
IF ((DEST & 4096) == 0)
        BCF 10,4
ELSE
        BSF 10,4
ENDIF
        ENDM
S@B     MACRO VARIN
IF((VARIN & 384) == 0)
IF(RAM_BANK == 1)
        BCF 3,5
ENDIF
IF(RAM_BANK == 2)
        BCF 3,6
ENDIF
IF(RAM_BANK == 3)
        BCF 3,5
        BCF 3,6
ENDIF
RAM_BANK = 0
ENDIF
IF((VARIN & 384) == 128)
IF(RAM_BANK == 0)
        BSF 3,5
ENDIF
IF(RAM_BANK == 2)
        BSF 3,5
        BCF 3,6
ENDIF
IF(RAM_BANK == 3)
        BCF 3,6
ENDIF
RAM_BANK = 1
ENDIF
IF((VARIN & 384) == 256)
IF(RAM_BANK == 0)
        BSF 3,6
ENDIF
IF(RAM_BANK == 1)
        BCF 3,5
        BSF 3,6
ENDIF
IF(RAM_BANK == 3)
        BCF 3,5
ENDIF
RAM_BANK = 2
ENDIF
IF((VARIN & 384) == 384)
IF(RAM_BANK == 0)
        BSF 3,5
        BSF 3,6
ENDIF
IF(RAM_BANK == 1)
        BSF 3,6
ENDIF
IF(RAM_BANK == 2)
        BSF 3,5
ENDIF
RAM_BANK = 3
ENDIF
        ENDM
R@B     MACRO
IF((RAM_BANK & 1) != 0)
        BCF 3,5
ENDIF
IF((RAM_BANK & 2) != 0)
        BCF 3,6
ENDIF
RAM_BANK = 0
        ENDM
JUMP MACRO PLABEL
    GOTO PLABEL
    ENDM
```

```
WREG_BYTE MACRO PBYTEOUT
    S@B PBYTEOUT
    MOVWF PBYTEOUT
    R@B
    ENDM
WREG_BIT MACRO PVAROUT,PBITOUT
    S@B PVAROUT
    ANDLW 1
    BTFSC STATUS,2
    BCF PVAROUT,PBITOUT
    BTFSS STATUS,2
    BSF PVAROUT,PBITOUT
    R@B
    ENDM
WREG_WORD MACRO PWORDOUT
    S@B PWORDOUT
    MOVWF PWORDOUT
    S@B PWORDOUT+1
    CLRF PWORDOUT+1
    R@B
    ENDM
WREG_DWORD MACRO PDWORDOUT
    S@B PDWORDOUT+3
    CLRF PDWORDOUT+3
    S@B PDWORDOUT+2
    CLRF PDWORDOUT+2
    S@B PDWORDOUT+1
    CLRF PDWORDOUT+1
    S@B PDWORDOUT
    MOVWF PDWORDOUT
    R@B
    ENDM
BYTE_WREG MACRO PBYTEIN
    S@B PBYTEIN
    MOVF PBYTEIN,W
    R@B
    ENDM
NUM_WREG MACRO PNUMIN
    MOVLW (PNUMIN & 255)
    ENDM
NUM_BYTE MACRO PNUMIN,PBYTEOUT
    S@B PBYTEOUT
IF(PNUMIN == 0)
    CLRF PBYTEOUT
ELSE
    MOVLW (PNUMIN & 255)
    MOVWF PBYTEOUT
ENDIF
    R@B
    ENDM
NUM_BIT MACRO PNUMIN,PVAROUT,PBITOUT
    S@B PVAROUT
IF((PNUMIN & 1) == 1)
    BSF PVAROUT,PBITOUT
ELSE
    BCF PVAROUT,PBITOUT
ENDIF
    R@B
    ENDM
NUM_WORD MACRO PNUMIN,PWORDOUT
IF((PNUMIN & 255) == 0)
    S@B PWORDOUT
    CLRF PWORDOUT
ELSE
    S@B PWORDOUT
    MOVLW LOW (PNUMIN)
    MOVWF PWORDOUT
ENDIF
IF(((PNUMIN >> 8) & 255) == 0)
    S@B PWORDOUT+1
    CLRF PWORDOUT+1
ELSE
    S@B PWORDOUT+1
    MOVLW HIGH (PNUMIN)
    MOVWF PWORDOUT+1
ENDIF
    R@B
    ENDM
NUM_DWORD MACRO PNUMIN,PDWORDOUT
IF ((PNUMIN >> 24 & 255) == 0)
    S@B PDWORDOUT+3
    CLRF PDWORDOUT+3
ELSE
    S@B PDWORDOUT+3
    MOVLW ((PNUMIN >> 24) & 255)
    MOVWF PDWORDOUT+3
ENDIF
IF( ((PNUMIN >> 16) & 255) == 0)
    S@B PDWORDOUT+2
    CLRF PDWORDOUT+2
ELSE
    S@B PDWORDOUT+2
    MOVLW ((PNUMIN >> 16) & 255)
    MOVWF PDWORDOUT+2
ENDIF
IF( ((PNUMIN >> 8) & 255) == 0)
    S@B PDWORDOUT+1
    CLRF PDWORDOUT+1
ELSE
    S@B PDWORDOUT+1
    MOVLW HIGH (PNUMIN)
    MOVWF PDWORDOUT+1
ENDIF
IF((PNUMIN & 255) == 0)
    S@B PDWORDOUT
    CLRF PDWORDOUT
ELSE
    S@B PDWORDOUT
    MOVLW LOW (PNUMIN)
    MOVWF PDWORDOUT
ENDIF
    R@B
    ENDM
BIT_WREG MACRO PVARIN,PBITIN
```

```
    S@B PVARIN
    CLRW
    BTFSC PVARIN,PBITIN
    MOVLW 1
    R@B
    ENDM
BIT_BYTE MACRO
PVARIN,PBITIN,PBYTEOUT
    S@B PVARIN
    CLRW
    BTFSC PVARIN,PBITIN
    MOVLW 1
    S@B PBYTEOUT
    MOVWF PBYTEOUT
    R@B
    ENDM
BIT_BIT MACRO
PVARIN,PBITIN,PVAROUT,PBITOUT
IF((PVARIN & 65408) == (PVAROUT &
65408))
    S@B PVAROUT
    BTFSC PVARIN,PBITIN
    BSF PVAROUT,PBITOUT
    BTFSS PVARIN,PBITIN
    BCF PVAROUT,PBITOUT
ELSE
    S@B PVARIN
    CLRDC
    BTFSC PVARIN,PBITIN
    SETDC
    S@B PVAROUT
    SKPNDC
    BSF PVAROUT,PBITOUT
    SKPDC
    BCF PVAROUT,PBITOUT
ENDIF
    ENDM
BIT_WORD MACRO
PVARIN,PBITIN,PWORDOUT
    S@B PWORDOUT+1
    CLRF PWORDOUT+1
    BIT_BYTE PVARIN,PBITIN,PWORDOUT
    ENDM
BIT_DWORD MACRO
PVARIN,PBITIN,PDWORDOUT
    S@B PDWORDOUT+3
    CLRF PDWORDOUT+3
    S@B PDWORDOUT+2
    CLRF PDWORDOUT+2
    S@B PDWORDOUT+1
    CLRF PDWORDOUT+1
    BIT_BYTE PVARIN,PBITIN,PDWORDOUT
    ENDM
WORD_WREG MACRO PWORDIN
    BYTE_WREG PWORDIN
    ENDM
WORD_BYTE MACRO PWORDIN,PBYTEOUT
    BYTE_BYTE PWORDIN,PBYTEOUT
    ENDM
WORD_BIT MACRO
PWORDIN,PVAROUT,PBITOUT
    BYTE_BIT PWORDIN, PVAROUT,
PBITOUT
    ENDM
WORD_WORD MACRO PWORDIN,PWORDOUT
    S@B PWORDIN+1
    MOVF PWORDIN+1,W
    S@B PWORDOUT+1
    MOVWF PWORDOUT+1
    BYTE_BYTE PWORDIN,PWORDOUT
    ENDM
WORD_DWORD MACRO PWORDIN,PDWORDOUT
    S@B PDWORDOUT+3
    CLRF PDWORDOUT+3
    S@B PDWORDOUT+2
    CLRF PDWORDOUT+2
    BYTE_BYTE PWORDIN+1,PDWORDOUT+1
    BYTE_BYTE PWORDIN,PDWORDOUT
    ENDM
BYTE_BYTE MACRO PBYTEIN,PBYTEOUT
    S@B PBYTEIN
    MOVF PBYTEIN,W
    S@B PBYTEOUT
    MOVWF PBYTEOUT
    R@B
    ENDM
BYTE_WORD MACRO PBYTEIN,PWORDOUT
    S@B PWORDOUT+1
    CLRF PWORDOUT+1
    BYTE_BYTE PBYTEIN,PWORDOUT
    ENDM
BYTE_DWORD MACRO PBYTEIN,PDWORDOUT
    S@B PDWORDOUT+3
    CLRF PDWORDOUT+3
    S@B PDWORDOUT+2
    CLRF PDWORDOUT+2
    S@B PDWORDOUT+1
    CLRF PDWORDOUT+1
    BYTE_BYTE PBYTEIN,PDWORDOUT
    ENDM
    BYTE_BIT MACRO
PBYTEIN,PVAROUT,PBITOUT
IF((PBYTEIN & 65408) == (PVAROUT &
65408))
    S@B PBYTEIN
    BTFSC PBYTEIN,0
    BSF PVAROUT,PBITOUT
    BTFSS PBYTEIN,0
    BCF PVAROUT,PBITOUT
ELSE
    S@B PBYTEIN
    RRF PBYTEIN,W
    S@B PVAROUT
    SKPNC
```

```
    BSF PVAROUT,PBITOUT
    SKPC
    BCF PVAROUT,PBITOUT
ENDIF
    R@B
    ENDM
DWORD_WREG MACRO PDWORDIN
    BYTE_WREG PDWORDIN
    ENDM
DWORD_BYTE MACRO PDWORDIN,PBYTEOUT
    BYTE_BYTE PDWORDIN,PBYTEOUT
    ENDM
DWORD_WORD MACRO PDWORDIN,PWORDOUT
    S@B PDWORDIN+1
    MOVF PDWORDIN+1,W
    S@B PWORDOUT+1
    MOVWF PWORDOUT+1
    BYTE_BYTE PDWORDIN,PWORDOUT
    ENDM
DWORD_DWORD MACRO PDWORDIN,PDWORDOUT
    BYTE_BYTE PDWORDIN+3,PDWORDOUT+3
    BYTE_BYTE PDWORDIN+2,PDWORDOUT+2
    BYTE_BYTE PDWORDIN+1,PDWORDOUT+1
    BYTE_BYTE PDWORDIN,PDWORDOUT
    ENDM
DWORD_BIT MACRO
PDWORDIN,PVAROUT,PBITOUT
    BYTE_BIT
PDWORDIN,PVAROUT,PBITOUT
    ENDM
VARIABLE CURRENT@PAGE = 0
VARIABLE PDESTINATION@PAGE = 0
FIND@PAGE MACRO PLABELIN
LOCAL CURRENT_ADDR = $
LOCAL DEST_ADDR = PLABELIN
IF((CURRENT_ADDR >= 0X1800) &&
(CURRENT_ADDR <= 0X2000))
    CURRENT@PAGE = 3
ENDIF
IF((CURRENT_ADDR >= 0X1000) &&
(CURRENT_ADDR <= 0X1800))
    CURRENT@PAGE = 2
ENDIF
IF((CURRENT_ADDR >= 0X0800) &&
(CURRENT_ADDR <= 0X1000))
    CURRENT@PAGE = 1
ENDIF
IF((CURRENT_ADDR >= 0) &&
(CURRENT_ADDR <= 0X0800))
    CURRENT@PAGE = 0
ENDIF
IF((DEST_ADDR >= 0X1800) &&
(DEST_ADDR <= 0X2000))
    PDESTINATION@PAGE = 3
ENDIF
IF((DEST_ADDR >= 0X1000) &&
(DEST_ADDR <= 0X1800))
    PDESTINATION@PAGE = 2
ENDIF
IF((DEST_ADDR >= 0X0800) &&
(DEST_ADDR <= 0X1000))
    PDESTINATION@PAGE = 1
ENDIF
IF((DEST_ADDR >= 0) && (DEST_ADDR <=
0X0800))
    PDESTINATION@PAGE = 0
ENDIF
    ENDM
NUM_FSR MACRO PNUMIN
    NUM_BYTE PNUMIN, FSR
IF (((PNUMIN >> 8) & 255) == 1)
    BSF STATUS,7
ELSE
    BCF STATUS,7
ENDIF
    ENDM
LABEL_WORD MACRO PLABELIN,PWORDOUT
    MOVLW LOW (PLABELIN)
    S@B PWORDOUT
    MOVWF PWORDOUT
    MOVLW HIGH (PLABELIN)
    S@B PWORDOUT
    MOVWF PWORDOUT+1
    R@B
    ENDM
BPF = 32
BPFH = 33
CSTR#CT = 34
GEN = 35
GEN3H = 36
GEN4 = 37
GEN4H = 38
GPR = 39
PP0 = 40
PP0H = 41
PP0HH = 42
PP0HHH = 43
PP1 = 44
PP1H = 45
PP2 = 46
PP2H = 47
PP2HH = 48
PP2HHH = 49
PP3 = 50
PP3H = 51
PP4 = 52
PP5 = 53
PP6 = 54
PP6H = 55
PPZ = 56
PPZH = 57
PPZHH = 58
PPZHHH = 59
RPFH = 60
```

```
SP#P9 = 61
FSRSAVE = 124
PSAVE = 125
SSAVE = 126
WSAVE = 127
DSDELAY = 62
DSDELAYH = 63
ICTR = 64
ICTRH = 65
ICTRHH = 66
ICTRHHH = 67
MN = 68
SC = 69
AHR = 70
AMN = 71
THR = 72
VARIABLE THR#0=72,THR#1=73,THR#2=74
TMN = 75
VARIABLE TMN#0=75,TMN#1=76,TMN#2=77
GCTR = 78
GSHFC = 79
GSHF = 80
GSHFL = 81
ATYPE = 82
DSMODE = 83
KEYTEMP = 84
#DEFINE __LCD_DTPORT PORTC
#DEFINE __LCD_RSPORT PORTD
#DEFINE __LCD_ENPORT PORTD
#DEFINE T0IF INTCON,2
#DEFINE T0IE INTCON,5
#DEFINE GIE INTCON,7
#DEFINE PORTB0_S PORTD,2
#DEFINE PORTB1_S PORTD,3
#DEFINE PORTB2_S PORTD,4
#DEFINE PORTB3_S PORTC,4
#DEFINE PORTB4_S PORTC,5
#DEFINE PORTB5_S PORTC,6
#DEFINE PORTB6_S PORTC,7
#DEFINE PORTB7_S PORTA,5
#DEFINE BUZZERS PORTA,4
#DEFINE __XTAL 4
#DEFINE __LCD_DTPIN 0
#DEFINE __LCD_RSPIN 0
#DEFINE __LCD_ENPIN 1
#DEFINE __LCD_LINES 2
#DEFINE __LCD_INTERFACE 4
PROTON#CODE#START
        ORG 0
        NOP
        MOVLW HIGH
(PROTON#MAIN#START)
        MOVWF 10
        GOTO PROTON#MAIN#START
        ORG 4
        MOVWF WSAVE
        MOVF 3,W
        CLRF 3
        MOVWF SSAVE
        MOVF 4,W
        MOVWF FSRSAVE
        MOVF 10,W
        MOVWF PSAVE
        MOVLW (MYCLOCKER >> 8) & 256
        MOVWF 10
        GOTO MYCLOCKER
T@GTB
        CLRF 41
T@GTW
        CLRF 45
T@GT
        MOVWF 44
        MOVLW 1
T@ST
        MOVWF 39
        MOVF 45,W
        SUBWF 41,W
        SKPZ
        GOTO $ + 3
        MOVF 44,W
        SUBWF 40,W
        MOVLW 4
        SKPNC
        MOVLW 1
        SKPNZ
        MOVLW 2
        ANDWF 39,W
        SKPZ
        MOVLW 1
        GOTO I@NT
LCD@CLS
        MOVLW 128
        MOVWF 33
CLS
        MOVLW 254
        CALL CH@SND
        MOVLW 1
        CALL CH@SND
        MOVLW 117
        MOVWF 41
        MOVLW 48
        GOTO DLUS@W
LCD@CUR
        CLRF 33
        BSF 33,7
CURSOR
        MOVWF 53
        TSTF 53
        SKPZ
        GOTO $ + 3
        CLRW
        GOTO $ + 2
        MOVLW 1
        SUBWF 53,F
```

```
        BCF 3,0
        RRF 35,F
        RRF 35,F
        RRF 35,F
        MOVLW 192
        ANDWF 35,W
        ADDLW 64
        ADDWF 53,F
        MOVLW 254
        CALL CH@SND
        MOVF 53,W
        GOTO CH@SND
LCD@CRS
        CLRF 33
        BSF 33,7
CURS@
        MOVWF 53
        MOVLW 254
        CALL CH@SND
        MOVF 53,W
        GOTO CH@SND
IN@DEC32
        CLRF 46
        CLRF 47
        CLRF 48
        CLRF 49
        BCF 39,5
IN@DECB32
        CALL CH@RCV
        ADDLW 211
        SKPNZ
        BSF 39,5
        ADDLW 45
        ADDLW 198
        SKPNC
        GOTO IN@DECB32
        ADDLW 10
        SKPC
        GOTO IN@DECB32
IN@DCL132
        MOVWF 44
        CLRF 43
        CLRF 42
        CLRF 41
        MOVLW 10
        MOVWF 40
        CALL AL@MUL
        MOVF 44,W
        ADDWF 40,W
        MOVWF 46
        MOVF 41,W
        MOVWF 47
        CLRW
        SKPNC
        MOVLW 1
        ADDWF 47,F
        MOVF 42,W
        MOVWF 48
        CLRW
        SKPNC
        MOVLW 1
        ADDWF 48,F
        MOVF 43,W
        MOVWF 49
        CLRW
        SKPNC
        MOVLW 1
        ADDWF 49,F
        DECF 38,F
        SKPNZ
        GOTO IN@BNF32
        CALL CH@RCV
        ADDLW 198
        SKPNC
        GOTO IN@BNF32
        ADDLW 10
        SKPNC
        GOTO IN@DCL132
IN@BNF32
        BTFSS 39,5
        GOTO $ + 12
        COMF 46,F
        COMF 47,F
        COMF 48,F
        COMF 49,F
        INCF 46,F
        SKPNZ
        INCF 47,F
        SKPNZ
        INCF 48,F
        SKPNZ
        INCF 49,F
        MOVF 46,W
        GOTO I@NT
OUT@DECB
        CLRF 38
OUT@DECC
        MOVWF 46
        CLRF 47
OUT@DEC
        BCF 32,3
        MOVF 38,W
        SKPNZ
        BSF 32,3
        MOVLW 5
        MOVWF 37
        MOVLW 39
        MOVWF 45
        MOVLW 16
        CALL D@DIG
        MOVLW 3
        MOVWF 45
        MOVLW 232
        CALL D@DIG
```

```
        CLRF 45
        MOVLW 100
        CALL D@DIG
        CLRF 45
        MOVLW 10
        CALL D@DIG
        MOVF 46,W
        GOTO SEND@IT
D@DIG
        MOVWF 44
        MOVF 47,W
        MOVWF 41
        MOVF 46,W
        MOVWF 40
        CALL D@VD
        MOVF 40,W
SEND@IT
        MOVWF 40
        DECF 37,F
        SKPNZ
        BCF 32,3
        MOVF 38,W
        SKPNZ
        GOTO $ + 4
        SUBWF 37,W
        SKPNC
        GOTO EX@SEND@IT
        MOVF 40,W
        SKPZ
        BCF 32,3
        BTFSC 32,3
        GOTO EX@SEND@IT
        ADDLW '0'
        GOTO CH@SND
EX@SEND@IT
        RETURN
PRINT
        MOVWF 51
        BCF PORTD,1
        BCF PORTD,0
        BSF 3,5
        BCF PORTD,1
        BCF PORTD,0
        MOVLW 240
        ANDWF PORTC,F
        BCF 3,5
        MOVF 51,W
        BTFSC 32,1
        GOTO PRT@1
        MOVLW 62
        MOVWF 41
        MOVLW 128
        CALL DLUS@W
        MOVLW 51
        MOVWF 50
        CALL PR@LP
        MOVLW 19
        MOVWF 41
        MOVLW 136
        CALL DLUS@W
        CALL PR@LP
        MOVLW 100
        CALL DL@US
        CALL PR@LP
        MOVLW 100
        CALL DL@US
        MOVLW 34
        MOVWF 50
        CALL PR@LP
        MOVLW 40
        CALL PRT@C
        MOVLW 12
        CALL PRT@C
        MOVLW 6
        CALL PRT@C
        BSF 32,1
        MOVF 51,W
        GOTO PRT@1
PRT@C
        BSF 32,0
PRT@1
        MOVWF 50
        BTFSS 32,0
        GOTO PRIT@SKP
        BCF PORTD,0
        SUBLW 3
        SKPC
        GOTO PRT@S
        CALL PRT@S
        MOVLW 7
        MOVWF 41
        MOVLW 208
        CALL DLUS@W
        BSF 3,0
        RETURN
PRIT@SKP
        BSF 32,0
        SUBLW 254
        SKPNZ
        GOTO PR@EX
        BSF PORTD,0
PRT@S
        SWAPF 50,F
        BTFSS 32,0
PR@LP
        BCF 32,0
        BSF PORTD,1
        MOVLW 240
        ANDWF PORTC,F
        MOVF 50,W
        ANDLW 15
        IORWF PORTC,F
        BCF PORTD,1
        SWAPF 50,F
```

```
        BTFSC 32,0
        GOTO PR@LP
        MOVLW 50
        CALL DL@US
PR@EX
        MOVF 51,W
        GOTO I@NT
CH@SND
        BTFSC 33,7
        GOTO PRINT
        RETURN
CH@RCV
        BTFSS 60,7
        RETURN
        MOVF 0,W
        INCF 4,F
        RETURN
C@STROUT
        MOVWF 37
C@STROUTW
        CLRF 34
C@STROUTLP
        MOVF 37,W
        BSF 3,6
        MOVWF 269
        BCF 3,6
        MOVF 38,W
        BSF 3,6
        MOVWF 271
        CALL C@READ
        SKPNZ
        GOTO I@NT
        CALL CH@SND
        INCF 37,F
        SKPNZ
        INCF 38,F
        INCFSZ 34,F
        GOTO C@STROUTLP
        GOTO I@NT
C@READ
        BSF 3,5
        BSF 3,6
        BSF 396,PP_EEPGD
        BSF 396,PP_RD
        NOP
        NOP
        BCF 3,5
        MOVF 268,W
        GOTO I@NT
DL@MS
        CLRF 45
DLY@W
        MOVWF 44
DLY@P
        MOVLW 255
        ADDWF 44,F
        SKPC
        ADDWF 45,F
        SKPC
        GOTO I@NT
        MOVLW 3
        MOVWF 41
        MOVLW 223
        CALL DLUS@W
        GOTO DLY@P
DL@US
        CLRF 41
DLUS@W
        ADDLW 232
        MOVWF 40
        COMF 41,F
        MOVLW 252
        SKPC
        GOTO $ + 4
        ADDWF 40,F
        SKPNC
        GOTO $ - 2
        ADDWF 40,F
        NOP
        INCFSZ 41,F
        GOTO $ - 6
        BTFSC 40,0
        GOTO $ + 1
        BTFSS 40,1
        GOTO $ + 3
        NOP
        GOTO $ + 1
        RETURN
D@VD
        CLRF 47
        CLRF 46
D@VD2
        MOVLW 16
        MOVWF 50
        RLF 41,W
        RLF 46,F
        RLF 47,F
        MOVF 44,W
        SUBWF 46,F
        MOVF 45,W
        SKPC
        INCFSZ 45,W
        SUBWF 47,F
        SKPNC
        GOTO $ + 8
        MOVF 44,W
        ADDWF 46,F
        MOVF 45,W
        SKPNC
        INCFSZ 45,W
        ADDWF 47,F
        BCF 3,0
        RLF 40,F
        RLF 41,F
```

```
        DECFSZ 50,F
        GOTO $ - 21
        MOVF 40,W
        GOTO I@NT
AL@MUL
        MOVLW 32
        MOVWF 50
        CLRF 56
        CLRF 57
        CLRF 58
        CLRF 59
MLT32@LP
        CLRC
        BTFSS 40,0
        GOTO MLT32@BR
        MOVF 46,W
        ADDWF 56,F
        MOVF 47,W
        SKPNC
        INCFSZ 47,W
        ADDWF 57,F
        MOVF 48,W
        SKPNC
        INCFSZ 48,W
        ADDWF 58,F
        MOVF 49,W
        SKPNC
        INCFSZ 49,W
        ADDWF 59,F
MLT32@BR
        RRF 59,F
        RRF 58,F
        RRF 57,F
        RRF 56,F
        RRF 43,F
        RRF 42,F
        RRF 41,F
        RRF 40,F
        DECFSZ 50,F
        GOTO MLT32@LP
        RETURN
I@NT
        BCF 3,7
I@NT2
        BCF 3,5
        BCF 3,6
        RETURN
VALDEC@32
        MOVWF 4
        CLRF 38
        MOVLW 128
        MOVWF 60
        GOTO IN@DEC32
PROTON#MAIN#START
        CLRF 32
F2_SOF EQU $ ; MITLEDXP.PRP
F2_EOF EQU $ ; MITLEDXP.PRP
F1_SOF EQU $ ; MITLEDXP.BAS
F1_000023 EQU $ ; IN [MITLEDXP.BAS]
ALL_DIGITAL=TRUE
        MOVLW 7
        BSF STATUS,5
RAM_BANK = 1
        MOVWF CMCON
        MOVWF ADCON1
F1_000065 EQU $ ; IN [MITLEDXP.BAS]
GOTO PREPROG
        BCF STATUS,5
RAM_BANK = 0
        F@JUMP PREPROG
MYCLOCKER
RAM_BANK = 0
F1_000069 EQU $ ; IN [MITLEDXP.BAS]
IF T0IF = 1 THEN
        SET@PAGE BC@LL2
        BTFSS INTCON,2
        GOTO BC@LL2
F1_000072 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = DSDELAY + 1
        INCF DSDELAY,F
        BTFSC STATUS,2
        INCF DSDELAYH,F
F1_000076 EQU $ ; IN [MITLEDXP.BAS]
IF ICTR > 488 THEN
        SET@PAGE BC@LL4
        BTFSC ICTRHHH,7
        GOTO BC@LL4
        MOVF ICTRHH,W
        IORWF ICTRHHH,W
        SET@PAGE CP@LB2
        BTFSS STATUS,2
        GOTO CP@LB2
        MOVLW 1
        SUBWF ICTRH,W
        SET@PAGE BC@LL4
        BTFSS STATUS,0
        GOTO BC@LL4
        SET@PAGE CP@LB2
        BTFSS STATUS,2
        GOTO CP@LB2
        MOVLW 233
        SUBWF ICTR,W
        SET@PAGE BC@LL4
        BTFSS STATUS,0
        GOTO BC@LL4
CP@LB2
F1_000079 EQU $ ; IN [MITLEDXP.BAS]
IF SC > 0 THEN
        MOVF SC,F
        SET@PAGE BC@LL6
        BTFSC STATUS,2
        GOTO BC@LL6
F1_000081 EQU $ ; IN [MITLEDXP.BAS]
SC = SC - 1
```

```
        DECF SC,F
        F@JUMP BC@LL7
BC@LL6
F1_000083 EQU $ ; IN [MITLEDXP.BAS]
ELSE
F1_000085 EQU $ ; IN [MITLEDXP.BAS]
IF MN > 0 THEN
        MOVF MN,F
        SET@PAGE BC@LL9
        BTFSC STATUS,2
        GOTO BC@LL9
F1_000087 EQU $ ; IN [MITLEDXP.BAS]
MN = MN - 1
        DECF MN,F
F1_000089 EQU $ ; IN [MITLEDXP.BAS]
SC = 59
        MOVLW 59
        MOVWF SC
        F@JUMP BC@LL10
BC@LL9
F1_000091 EQU $ ; IN [MITLEDXP.BAS]
ELSE
F1_000093 EQU $ ; IN [MITLEDXP.BAS]
MN = 0
        CLRF MN
F1_000095 EQU $ ; IN [MITLEDXP.BAS]
SC = 0
        CLRF SC
F1_000098 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL10
F1_000101 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL7
F1_000103 EQU $ ; IN [MITLEDXP.BAS]
ICTR = 0
        CLRF ICTRHHH
        CLRF ICTRHH
        CLRF ICTRH
        CLRF ICTR
        F@JUMP BC@LL11
BC@LL4
F1_000105 EQU $ ; IN [MITLEDXP.BAS]
ELSE
F1_000107 EQU $ ; IN [MITLEDXP.BAS]
ICTR = ICTR + 1
        MOVLW 1
        ADDWF ICTR,F
        CLRW
        BTFSC STATUS,0
        MOVLW 1
        ADDWF ICTRH,F
        CLRW
        BTFSC STATUS,0
        MOVLW 1
        ADDWF ICTRHH,F
        CLRW
        BTFSC STATUS,0
        MOVLW 1
        ADDWF ICTRHHH,F
F1_000109 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL11
F1_000115 EQU $ ; IN [MITLEDXP.BAS]
IF GSHFC < GSHFL THEN
        MOVF GSHFL,W
        SUBWF GSHFC,W
        SET@PAGE BC@LL13
        BTFSC STATUS,0
        GOTO BC@LL13
F1_000118 EQU $ ; IN [MITLEDXP.BAS]
GSHFC = GSHFC + 1
        INCF GSHFC,F
        F@JUMP BC@LL14
BC@LL13
F1_000121 EQU $ ; IN [MITLEDXP.BAS]
ELSE
F1_000123 EQU $ ; IN [MITLEDXP.BAS]
GSHFC = 0
        CLRF GSHFC
F1_000125 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL14
F1_000127 EQU $ ; IN [MITLEDXP.BAS]
GSHF = GSHFC
        MOVF GSHFC,W
        MOVWF GSHF
F1_000128 EQU $ ; IN [MITLEDXP.BAS]
GSHF = GSHF << 5
        SWAPF GSHF,W
        ANDLW 240
        MOVWF GSHF
        BCF STATUS,0
        RLF GSHF,F
F1_000133 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD & $1F
        MOVLW 31
        ANDWF PORTD,F
F1_000135 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD | GSHF
        MOVF GSHF,W
        IORWF PORTD,F
F1_000154 EQU $ ; IN [MITLEDXP.BAS]
T0IF=0
        BCF INTCON,2
F1_000156 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL2
F1_000158 EQU $ ; IN [MITLEDXP.BAS]
CONTEXT RESTORE
        MOVF PSAVE,W
        MOVWF PCLATH
        MOVF FSRSAVE,W
        MOVWF FSR
```

```
        MOVF SSAVE,W
        MOVWF STATUS
        SWAPF WSAVE,F
        SWAPF WSAVE,W
        RETFIE
PREPROG
F1_000162 EQU $ ; IN [MITLEDXP.BAS]
TRISA = $0F
        MOVLW 15
        BSF STATUS,5
RAM_BANK = 1
        MOVWF TRISA
F1_000163 EQU $ ; IN [MITLEDXP.BAS]
TRISB = $00
        CLRF TRISB
F1_000164 EQU $ ; IN [MITLEDXP.BAS]
TRISC = $00
        CLRF TRISC
F1_000165 EQU $ ; IN [MITLEDXP.BAS]
TRISD = $00
        CLRF TRISD
F1_000166 EQU $ ; IN [MITLEDXP.BAS]
TRISE = $00
        CLRF TRISE
F1_000168 EQU $ ; IN [MITLEDXP.BAS]
OPTION_REG = $02
        MOVLW 2
        MOVWF OPTION_REG
F1_000169 EQU $ ; IN [MITLEDXP.BAS]
TMR0 = 0
        BCF STATUS,5
RAM_BANK = 0
        CLRF TMR0
F1_000170 EQU $ ; IN [MITLEDXP.BAS]
T0IE = 0
        BCF INTCON,5
F1_000171 EQU $ ; IN [MITLEDXP.BAS]
GIE=0
        BCF INTCON,7
F1_000175 EQU $ ; IN [MITLEDXP.BAS]
DELAYMS 500
        MOVLW 1
        MOVWF PP1H
        MOVLW 244
        F@CALL DLY@W
F1_000177 EQU $ ; IN [MITLEDXP.BAS]
PORTD = $00
        CLRF PORTD
F1_000179 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $00
        CLRF PORTB
F1_000181 EQU $ ; IN [MITLEDXP.BAS]
GSHFL = 0
        CLRF GSHFL
F1_000185 EQU $ ; IN [MITLEDXP.BAS]
TMR0 = 0
        CLRF TMR0
F1_000186 EQU $ ; IN [MITLEDXP.BAS]
T0IE = 1
        BSF INTCON,5
F1_000187 EQU $ ; IN [MITLEDXP.BAS]
GIE = 1
        BSF INTCON,7
STARTHERE
F1_000194 EQU $ ; IN [MITLEDXP.BAS]
GOSUB INITLEDS
        F@CALL INITLEDS
F1_000196 EQU $ ; IN [MITLEDXP.BAS]
GOSUB SETPCBSIZE
        F@CALL SETPCBSIZE
F1_000198 EQU $ ; IN [MITLEDXP.BAS]
CLS
        F@CALL LCD@CLS
F1_000199 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 1,1,"TIME:"
        MOVLW 128
        MOVWF BPFH
        F@CALL LCD@CRS
        MOVLW 84
        F@CALL PRINT
        MOVLW 73
        F@CALL PRINT
        MOVLW 77
        F@CALL PRINT
        MOVLW 69
        F@CALL PRINT
        MOVLW 58
        F@CALL PRINT
F1_000201 EQU $ ; IN [MITLEDXP.BAS]
WHILE MN > 0 OR SC > 0
BC@LL15
        MOVF MN,W
        MOVWF PP0
        MOVLW 0
        F@CALL T@GTB
        MOVWF SP#P9
        MOVF SC,W
        MOVWF PP0
        MOVLW 0
        F@CALL T@GTB
        IORWF SP#P9,F
        SET@PAGE BC@LL16
        BTFSC STATUS,2
        GOTO BC@LL16
F1_000203 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 2,1,DEC2 MN,":",DEC2 SC
        MOVLW 128
        MOVWF BPFH
        MOVLW 192
        F@CALL LCD@CRS
        MOVLW 2
        MOVWF GEN4H
        MOVF MN,W
        F@CALL OUT@DECC
```

```
        MOVLW 58
        F@CALL PRINT
        MOVLW 2
        MOVWF GEN4H
        MOVF SC,W
        F@CALL OUT@DECC
F1_000205 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL15
BC@LL16
F1_000207 EQU $ ; IN [MITLEDXP.BAS]
CLS
        F@CALL LCD@CLS
F1_000208 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 1,1,"EXPOSURE"
        MOVLW 128
        MOVWF BPFH
        F@CALL LCD@CRS
        MOVLW HIGH (STR@LB1)
        MOVWF GEN4H
        MOVLW LOW (STR@LB1)
        F@CALL C@STROUT
F1_000209 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 2,1,"COMPLETED."
        MOVLW 128
        MOVWF BPFH
        MOVLW 192
        F@CALL LCD@CRS
        MOVLW HIGH (STR@LB2)
        MOVWF GEN4H
        MOVLW LOW (STR@LB2)
        F@CALL C@STROUT
F1_000211 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 0
        BCF PORTA,4
F1_000212 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000213 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 1464
BC@LL17
        MOVLW 5
        SUBWF DSDELAYH,W
        MOVLW 184
        BTFSC STATUS,2
        SUBWF DSDELAY,W
        SET@PAGE BC@LL18
        BTFSC STATUS,0
        GOTO BC@LL18
F1_000214 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL17
BC@LL18
F1_000215 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 1
        BSF PORTA,4
F1_000221 EQU $ ; IN [MITLEDXP.BAS]
GOTO STARTHERE
        F@JUMP STARTHERE
INITLEDS
F1_000225 EQU $ ; IN [MITLEDXP.BAS]
GSHFL = 0
        CLRF GSHFL
F1_000226 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $00
        CLRF PORTB
F1_000227 EQU $ ; IN [MITLEDXP.BAS]
PORTB0_S = 0
        BCF PORTD,2
F1_000228 EQU $ ; IN [MITLEDXP.BAS]
PORTB1_S = 0
        BCF PORTD,3
F1_000229 EQU $ ; IN [MITLEDXP.BAS]
PORTB2_S = 0
        BCF PORTD,4
F1_000230 EQU $ ; IN [MITLEDXP.BAS]
PORTB3_S = 0
        BCF PORTC,4
F1_000231 EQU $ ; IN [MITLEDXP.BAS]
PORTB4_S = 0
        BCF PORTC,5
F1_000232 EQU $ ; IN [MITLEDXP.BAS]
PORTB5_S = 0
        BCF PORTC,6
F1_000233 EQU $ ; IN [MITLEDXP.BAS]
PORTB6_S = 0
        BCF PORTC,7
F1_000234 EQU $ ; IN [MITLEDXP.BAS]
PORTB7_S = 0
        BCF PORTA,5
F1_000236 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 1
        BSF PORTA,4
F1_000238 EQU $ ; IN [MITLEDXP.BAS]
RETURN
        RETURN
LEDLOOKUP
F1_000242 EQU $ ; IN [MITLEDXP.BAS]
SELECT CASE KEYTEMP
F1_000244 EQU $ ; IN [MITLEDXP.BAS]
CASE "1"
        MOVLW 49
        SUBWF KEYTEMP,W
        SET@PAGE BC@LL20
        BTFSS STATUS,2
        GOTO BC@LL20
F1_000246 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $01
        MOVLW 1
        MOVWF PORTB
        F@JUMP BC@LL19
BC@LL20
```

```
F1_000248 EQU $ ; IN [MITLEDXP.BAS]
CASE "2"
        MOVLW 50
        SUBWF KEYTEMP,W
        SET@PAGE BC@LL23
        BTFSS STATUS,2
        GOTO BC@LL23
F1_000250 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $03
        MOVLW 3
        MOVWF PORTB
        F@JUMP BC@LL19
BC@LL23
F1_000252 EQU $ ; IN [MITLEDXP.BAS]
CASE "3"
        MOVLW 51
        SUBWF KEYTEMP,W
        SET@PAGE BC@LL25
        BTFSS STATUS,2
        GOTO BC@LL25
F1_000254 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $07
        MOVLW 7
        MOVWF PORTB
        F@JUMP BC@LL19
BC@LL25
F1_000256 EQU $ ; IN [MITLEDXP.BAS]
CASE "4"
        MOVLW 52
        SUBWF KEYTEMP,W
        SET@PAGE BC@LL27
        BTFSS STATUS,2
        GOTO BC@LL27
F1_000258 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $0F
        MOVLW 15
        MOVWF PORTB
        F@JUMP BC@LL19
BC@LL27
F1_000260 EQU $ ; IN [MITLEDXP.BAS]
CASE "5"
        MOVLW 53
        SUBWF KEYTEMP,W
        SET@PAGE BC@LL29
        BTFSS STATUS,2
        GOTO BC@LL29
F1_000262 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $1F
        MOVLW 31
        MOVWF PORTB
        F@JUMP BC@LL19
BC@LL29
F1_000264 EQU $ ; IN [MITLEDXP.BAS]
CASE "6"
        MOVLW 54
        SUBWF KEYTEMP,W
        SET@PAGE BC@LL31
        BTFSS STATUS,2
        GOTO BC@LL31
F1_000266 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $3F
        MOVLW 63
        MOVWF PORTB
        F@JUMP BC@LL19
BC@LL31
F1_000268 EQU $ ; IN [MITLEDXP.BAS]
CASE "7"
        MOVLW 55
        SUBWF KEYTEMP,W
        SET@PAGE BC@LL33
        BTFSS STATUS,2
        GOTO BC@LL33
F1_000270 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $7F
        MOVLW 127
        MOVWF PORTB
        F@JUMP BC@LL19
BC@LL33
F1_000272 EQU $ ; IN [MITLEDXP.BAS]
CASE "8"
        MOVLW 56
        SUBWF KEYTEMP,W
        SET@PAGE BC@LL35
        BTFSS STATUS,2
        GOTO BC@LL35
F1_000274 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $FF
        MOVLW 255
        MOVWF PORTB
F1_000278 EQU $ ; IN [MITLEDXP.BAS]
ENDSELECT
BC@LL35
BC@LL19
F1_000282 EQU $ ; IN [MITLEDXP.BAS]
RETURN
        RETURN
MYNUMKEYPAD
F1_000286 EQU $ ; IN [MITLEDXP.BAS]
WHILE 1 = 1
BC@LL36
F1_000288 EQU $ ; IN [MITLEDXP.BAS]
PORTE.0=0
        BCF PORTE,0
F1_000289 EQU $ ; IN [MITLEDXP.BAS]
PORTE.1=1
        BSF PORTE,1
F1_000290 EQU $ ; IN [MITLEDXP.BAS]
PORTE.2=1
        BSF PORTE,2
F1_000292 EQU $ ; IN [MITLEDXP.BAS]
IF PORTA.0=0 THEN
        SET@PAGE BC@LL39
        BTFSC PORTA,0
        GOTO BC@LL39
```

```
F1_000293 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000294 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL40
        MOVF DSDELAYH,F
        SET@PAGE BC@LL41
        BTFSS STATUS,2
        GOTO BC@LL41
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL41
        BTFSC STATUS,0
        GOTO BC@LL41
F1_000295 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL40
BC@LL41
F1_000296 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 0
        BCF PORTA,4
F1_000297 EQU $ ; IN [MITLEDXP.BAS]
WHILE PORTA.0=0
BC@LL42
        SET@PAGE BC@LL43
        BTFSC PORTA,0
        GOTO BC@LL43
F1_000298 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL42
BC@LL43
F1_000299 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 1
        BSF PORTA,4
F1_000300 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000301 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL44
        MOVF DSDELAYH,F
        SET@PAGE BC@LL45
        BTFSS STATUS,2
        GOTO BC@LL45
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL45
        BTFSC STATUS,0
        GOTO BC@LL45
F1_000302 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL44
BC@LL45
F1_000303 EQU $ ; IN [MITLEDXP.BAS]
KEYTEMP="1"
        MOVLW 49
        MOVWF KEYTEMP
F1_000304 EQU $ ; IN [MITLEDXP.BAS]
BREAK
        F@JUMP BC@LL37
        F@JUMP BC@LL38
F1_000306 EQU $ ; IN [MITLEDXP.BAS]
ELSEIF PORTA.1=0 THEN
BC@LL39
        SET@PAGE BC@LL46
        BTFSC PORTA,1
        GOTO BC@LL46
F1_000308 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000309 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL47
        MOVF DSDELAYH,F
        SET@PAGE BC@LL48
        BTFSS STATUS,2
        GOTO BC@LL48
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL48
        BTFSC STATUS,0
        GOTO BC@LL48
F1_000310 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL47
BC@LL48
F1_000311 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 0
        BCF PORTA,4
F1_000312 EQU $ ; IN [MITLEDXP.BAS]
WHILE PORTA.1=0
BC@LL49
        SET@PAGE BC@LL50
        BTFSC PORTA,1
        GOTO BC@LL50
F1_000313 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL49
BC@LL50
F1_000314 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 1
        BSF PORTA,4
F1_000315 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000316 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL51
```

```
        MOVF DSDELAYH,F
        SET@PAGE BC@LL52
        BTFSS STATUS,2
        GOTO BC@LL52
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL52
        BTFSC STATUS,0
        GOTO BC@LL52
F1_000317 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL51
BC@LL52
F1_000318 EQU $ ; IN [MITLEDXP.BAS]
KEYTEMP="4"
        MOVLW 52
        MOVWF KEYTEMP
F1_000319 EQU $ ; IN [MITLEDXP.BAS]
BREAK
        F@JUMP BC@LL37
        F@JUMP BC@LL38
F1_000321 EQU $ ; IN [MITLEDXP.BAS]
ELSEIF PORTA.2=0 THEN
BC@LL46
        SET@PAGE BC@LL53
        BTFSC PORTA,2
        GOTO BC@LL53
F1_000322 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000323 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL54
        MOVF DSDELAYH,F
        SET@PAGE BC@LL55
        BTFSS STATUS,2
        GOTO BC@LL55
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL55
        BTFSC STATUS,0
        GOTO BC@LL55
F1_000324 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL54
BC@LL55
F1_000325 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 0
        BCF PORTA,4
F1_000326 EQU $ ; IN [MITLEDXP.BAS]
WHILE PORTA.2=0
BC@LL56
        SET@PAGE BC@LL57
        BTFSC PORTA,2
        GOTO BC@LL57
F1_000327 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL56
BC@LL57
F1_000328 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 1
        BSF PORTA,4
F1_000329 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000330 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL58
        MOVF DSDELAYH,F
        SET@PAGE BC@LL59
        BTFSS STATUS,2
        GOTO BC@LL59
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL59
        BTFSC STATUS,0
        GOTO BC@LL59
F1_000331 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL58
BC@LL59
F1_000332 EQU $ ; IN [MITLEDXP.BAS]
KEYTEMP="7"
        MOVLW 55
        MOVWF KEYTEMP
F1_000333 EQU $ ; IN [MITLEDXP.BAS]
BREAK
        F@JUMP BC@LL37
        F@JUMP BC@LL38
F1_000336 EQU $ ; IN [MITLEDXP.BAS]
ELSEIF PORTA.3=0 THEN
BC@LL53
        SET@PAGE BC@LL60
        BTFSC PORTA,3
        GOTO BC@LL60
F1_000337 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000338 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL61
        MOVF DSDELAYH,F
        SET@PAGE BC@LL62
        BTFSS STATUS,2
        GOTO BC@LL62
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL62
        BTFSC STATUS,0
        GOTO BC@LL62
```

```
F1_000339 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL61
BC@LL62
F1_000340 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 0
        BCF PORTA,4
F1_000341 EQU $ ; IN [MITLEDXP.BAS]
WHILE PORTA.3=0
BC@LL63
        SET@PAGE BC@LL64
        BTFSC PORTA,3
        GOTO BC@LL64
F1_000342 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL63
BC@LL64
F1_000343 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 1
        BSF PORTA,4
F1_000344 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000345 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL65
        MOVF DSDELAYH,F
        SET@PAGE BC@LL66
        BTFSS STATUS,2
        GOTO BC@LL66
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL66
        BTFSC STATUS,0
        GOTO BC@LL66
F1_000346 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL65
BC@LL66
F1_000347 EQU $ ; IN [MITLEDXP.BAS]
KEYTEMP="*"
        MOVLW 42
        MOVWF KEYTEMP
F1_000348 EQU $ ; IN [MITLEDXP.BAS]
BREAK
        F@JUMP BC@LL37
F1_000349 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL60
BC@LL38
F1_000352 EQU $ ; IN [MITLEDXP.BAS]
PORTE.0=1
        BSF PORTE,0
F1_000353 EQU $ ; IN [MITLEDXP.BAS]
PORTE.1=0
        BCF PORTE,1
F1_000354 EQU $ ; IN [MITLEDXP.BAS]
PORTE.2=1
        BSF PORTE,2
F1_000356 EQU $ ; IN [MITLEDXP.BAS]
IF PORTA.0=0 THEN
        SET@PAGE BC@LL68
        BTFSC PORTA,0
        GOTO BC@LL68
F1_000357 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000358 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL69
        MOVF DSDELAYH,F
        SET@PAGE BC@LL70
        BTFSS STATUS,2
        GOTO BC@LL70
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL70
        BTFSC STATUS,0
        GOTO BC@LL70
F1_000359 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL69
BC@LL70
F1_000360 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 0
        BCF PORTA,4
F1_000361 EQU $ ; IN [MITLEDXP.BAS]
WHILE PORTA.0=0
BC@LL71
        SET@PAGE BC@LL72
        BTFSC PORTA,0
        GOTO BC@LL72
F1_000362 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL71
BC@LL72
F1_000363 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 1
        BSF PORTA,4
F1_000364 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000365 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL73
        MOVF DSDELAYH,F
        SET@PAGE BC@LL74
        BTFSS STATUS,2
        GOTO BC@LL74
        MOVLW 24
        SUBWF DSDELAY,W
```

```
        SET@PAGE BC@LL74
        BTFSC STATUS,0
        GOTO BC@LL74
F1_000366 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL73
BC@LL74
F1_000367 EQU $ ; IN [MITLEDXP.BAS]
KEYTEMP="2"
        MOVLW 50
        MOVWF KEYTEMP
F1_000368 EQU $ ; IN [MITLEDXP.BAS]
BREAK
        F@JUMP BC@LL37
        F@JUMP BC@LL67
F1_000370 EQU $ ; IN [MITLEDXP.BAS]
ELSEIF PORTA.1=0 THEN
BC@LL68
        SET@PAGE BC@LL75
        BTFSC PORTA,1
        GOTO BC@LL75
F1_000372 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000373 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL76
        MOVF DSDELAYH,F
        SET@PAGE BC@LL77
        BTFSS STATUS,2
        GOTO BC@LL77
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL77
        BTFSC STATUS,0
        GOTO BC@LL77
F1_000374 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL76
BC@LL77
F1_000375 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 0
        BCF PORTA,4
F1_000376 EQU $ ; IN [MITLEDXP.BAS]
WHILE PORTA.1=0
BC@LL78
        SET@PAGE BC@LL79
        BTFSC PORTA,1
        GOTO BC@LL79
F1_000377 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL78
BC@LL79
F1_000378 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 1
        BSF PORTA,4
F1_000379 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000380 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL80
        MOVF DSDELAYH,F
        SET@PAGE BC@LL81
        BTFSS STATUS,2
        GOTO BC@LL81
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL81
        BTFSC STATUS,0
        GOTO BC@LL81
F1_000381 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL80
BC@LL81
F1_000382 EQU $ ; IN [MITLEDXP.BAS]
KEYTEMP="5"
        MOVLW 53
        MOVWF KEYTEMP
F1_000383 EQU $ ; IN [MITLEDXP.BAS]
BREAK
        F@JUMP BC@LL37
        F@JUMP BC@LL67
F1_000385 EQU $ ; IN [MITLEDXP.BAS]
ELSEIF PORTA.2=0 THEN
BC@LL75
        SET@PAGE BC@LL82
        BTFSC PORTA,2
        GOTO BC@LL82
F1_000386 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000387 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL83
        MOVF DSDELAYH,F
        SET@PAGE BC@LL84
        BTFSS STATUS,2
        GOTO BC@LL84
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL84
        BTFSC STATUS,0
        GOTO BC@LL84
F1_000388 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL83
BC@LL84
F1_000389 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 0
        BCF PORTA,4
```

```
F1_000390 EQU $ ; IN [MITLEDXP.BAS]
WHILE PORTA.2=0
BC@LL85
        SET@PAGE BC@LL86
        BTFSC PORTA,2
        GOTO BC@LL86
F1_000391 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL85
BC@LL86
F1_000392 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 1
        BSF PORTA,4
F1_000393 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000394 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL87
        MOVF DSDELAYH,F
        SET@PAGE BC@LL88
        BTFSS STATUS,2
        GOTO BC@LL88
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL88
        BTFSC STATUS,0
        GOTO BC@LL88
F1_000395 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL87
BC@LL88
F1_000396 EQU $ ; IN [MITLEDXP.BAS]
KEYTEMP="8"
        MOVLW 56
        MOVWF KEYTEMP
F1_000397 EQU $ ; IN [MITLEDXP.BAS]
BREAK
        F@JUMP BC@LL37
        F@JUMP BC@LL67
F1_000400 EQU $ ; IN [MITLEDXP.BAS]
ELSEIF PORTA.3=0 THEN
BC@LL82
        SET@PAGE BC@LL89
        BTFSC PORTA,3
        GOTO BC@LL89
F1_000401 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000402 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL90
        MOVF DSDELAYH,F
        SET@PAGE BC@LL91
        BTFSS STATUS,2
        GOTO BC@LL91
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL91
        BTFSC STATUS,0
        GOTO BC@LL91
F1_000403 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL90
BC@LL91
F1_000404 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 0
        BCF PORTA,4
F1_000405 EQU $ ; IN [MITLEDXP.BAS]
WHILE PORTA.3=0
BC@LL92
        SET@PAGE BC@LL93
        BTFSC PORTA,3
        GOTO BC@LL93
F1_000406 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL92
BC@LL93
F1_000407 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 1
        BSF PORTA,4
F1_000408 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000409 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL94
        MOVF DSDELAYH,F
        SET@PAGE BC@LL95
        BTFSS STATUS,2
        GOTO BC@LL95
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL95
        BTFSC STATUS,0
        GOTO BC@LL95
F1_000410 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL94
BC@LL95
F1_000411 EQU $ ; IN [MITLEDXP.BAS]
KEYTEMP="0"
        MOVLW 48
        MOVWF KEYTEMP
F1_000412 EQU $ ; IN [MITLEDXP.BAS]
BREAK
        F@JUMP BC@LL37
F1_000413 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL89
BC@LL67
```

```
F1_000415 EQU $ ; IN [MITLEDXP.BAS]
PORTE.0=1
        BSF PORTE,0
F1_000416 EQU $ ; IN [MITLEDXP.BAS]
PORTE.1=1
        BSF PORTE,1
F1_000417 EQU $ ; IN [MITLEDXP.BAS]
PORTE.2=0
        BCF PORTE,2
F1_000418 EQU $ ; IN [MITLEDXP.BAS]
IF PORTA.0=0 THEN
        SET@PAGE BC@LL97
        BTFSC PORTA,0
        GOTO BC@LL97
F1_000419 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000420 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL98
        MOVF DSDELAYH,F
        SET@PAGE BC@LL99
        BTFSS STATUS,2
        GOTO BC@LL99
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL99
        BTFSC STATUS,0
        GOTO BC@LL99
F1_000421 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL98
BC@LL99
F1_000422 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 0
        BCF PORTA,4
F1_000423 EQU $ ; IN [MITLEDXP.BAS]
WHILE PORTA.0=0
BC@LL100
        SET@PAGE BC@LL101
        BTFSC PORTA,0
        GOTO BC@LL101
F1_000424 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL100
BC@LL101
F1_000425 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 1
        BSF PORTA,4
F1_000426 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000427 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL102
        MOVF DSDELAYH,F
        SET@PAGE BC@LL103
        BTFSS STATUS,2
        GOTO BC@LL103
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL103
        BTFSC STATUS,0
        GOTO BC@LL103
F1_000428 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL102
BC@LL103
F1_000429 EQU $ ; IN [MITLEDXP.BAS]
KEYTEMP="3"
        MOVLW 51
        MOVWF KEYTEMP
F1_000430 EQU $ ; IN [MITLEDXP.BAS]
BREAK
        F@JUMP BC@LL37
        F@JUMP BC@LL96
F1_000432 EQU $ ; IN [MITLEDXP.BAS]
ELSEIF PORTA.1=0 THEN
BC@LL97
        SET@PAGE BC@LL104
        BTFSC PORTA,1
        GOTO BC@LL104
F1_000434 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000435 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL105
        MOVF DSDELAYH,F
        SET@PAGE BC@LL106
        BTFSS STATUS,2
        GOTO BC@LL106
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL106
        BTFSC STATUS,0
        GOTO BC@LL106
F1_000436 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL105
BC@LL106
F1_000437 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 0
        BCF PORTA,4
F1_000438 EQU $ ; IN [MITLEDXP.BAS]
WHILE PORTA.1=0
BC@LL107
        SET@PAGE BC@LL108
        BTFSC PORTA,1
        GOTO BC@LL108
```

```
F1_000439 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL107
BC@LL108
F1_000440 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 1
        BSF PORTA,4
F1_000441 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000442 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL109
        MOVF DSDELAYH,F
        SET@PAGE BC@LL110
        BTFSS STATUS,2
        GOTO BC@LL110
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL110
        BTFSC STATUS,0
        GOTO BC@LL110
F1_000443 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL109
BC@LL110
F1_000444 EQU $ ; IN [MITLEDXP.BAS]
KEYTEMP="6"
        MOVLW 54
        MOVWF KEYTEMP
F1_000445 EQU $ ; IN [MITLEDXP.BAS]
BREAK
        F@JUMP BC@LL37
        F@JUMP BC@LL96
F1_000447 EQU $ ; IN [MITLEDXP.BAS]
ELSEIF PORTA.2=0 THEN
BC@LL104
        SET@PAGE BC@LL111
        BTFSC PORTA,2
        GOTO BC@LL111
F1_000448 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000449 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL112
        MOVF DSDELAYH,F
        SET@PAGE BC@LL113
        BTFSS STATUS,2
        GOTO BC@LL113
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL113
        BTFSC STATUS,0
        GOTO BC@LL113
F1_000450 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL112
BC@LL113
F1_000451 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 0
        BCF PORTA,4
F1_000452 EQU $ ; IN [MITLEDXP.BAS]
WHILE PORTA.2=0
BC@LL114
        SET@PAGE BC@LL115
        BTFSC PORTA,2
        GOTO BC@LL115
F1_000453 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL114
BC@LL115
F1_000454 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 1
        BSF PORTA,4
F1_000455 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000456 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL116
        MOVF DSDELAYH,F
        SET@PAGE BC@LL117
        BTFSS STATUS,2
        GOTO BC@LL117
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL117
        BTFSC STATUS,0
        GOTO BC@LL117
F1_000457 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL116
BC@LL117
F1_000458 EQU $ ; IN [MITLEDXP.BAS]
KEYTEMP="9"
        MOVLW 57
        MOVWF KEYTEMP
F1_000459 EQU $ ; IN [MITLEDXP.BAS]
BREAK
        F@JUMP BC@LL37
        F@JUMP BC@LL96
F1_000462 EQU $ ; IN [MITLEDXP.BAS]
ELSEIF PORTA.3=0 THEN
BC@LL111
        SET@PAGE BC@LL118
        BTFSC PORTA,3
        GOTO BC@LL118
F1_000463 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
```

```
        CLRF DSDELAYH
F1_000464 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL119
        MOVF DSDELAYH,F
        SET@PAGE BC@LL120
        BTFSS STATUS,2
        GOTO BC@LL120
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL120
        BTFSC STATUS,0
        GOTO BC@LL120
F1_000465 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL119
BC@LL120
F1_000466 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 0
        BCF PORTA,4
F1_000467 EQU $ ; IN [MITLEDXP.BAS]
WHILE PORTA.3=0
BC@LL121
        SET@PAGE BC@LL122
        BTFSC PORTA,3
        GOTO BC@LL122
F1_000468 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL121
BC@LL122
F1_000469 EQU $ ; IN [MITLEDXP.BAS]
BUZZERS = 1
        BSF PORTA,4
F1_000470 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000471 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 24
BC@LL123
        MOVF DSDELAYH,F
        SET@PAGE BC@LL124
        BTFSS STATUS,2
        GOTO BC@LL124
        MOVLW 24
        SUBWF DSDELAY,W
        SET@PAGE BC@LL124
        BTFSC STATUS,0
        GOTO BC@LL124
F1_000472 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL123
BC@LL124
F1_000473 EQU $ ; IN [MITLEDXP.BAS]
KEYTEMP="#"
        MOVLW 35
        MOVWF KEYTEMP
F1_000474 EQU $ ; IN [MITLEDXP.BAS]
BREAK
        F@JUMP BC@LL37
F1_000475 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL118
BC@LL96
F1_000478 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL36
BC@LL37
F1_000480 EQU $ ; IN [MITLEDXP.BAS]
RETURN
        RETURN
SETPCBSIZE
STSTART
F1_000487 EQU $ ; IN [MITLEDXP.BAS]
CLS
        F@CALL LCD@CLS
F1_000488 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 1,1,"SET PCB SIZE"
        MOVLW 128
        MOVWF BPFH
        F@CALL LCD@CRS
        MOVLW HIGH (STR@LB3)
        MOVWF GEN4H
        MOVLW LOW (STR@LB3)
        F@CALL C@STROUT
F1_000489 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 2,1,"00 X 00 INCH"
        MOVLW 128
        MOVWF BPFH
        MOVLW 192
        F@CALL LCD@CRS
        MOVLW HIGH (STR@LB4)
        MOVWF GEN4H
        MOVLW LOW (STR@LB4)
        F@CALL C@STROUT
F1_000492 EQU $ ; IN [MITLEDXP.BAS]
FOR GCTR = 0 TO 2
        CLRF GCTR
FR@LB126
        MOVLW 3
        SUBWF GCTR,W
        SET@PAGE NX@LB127
        BTFSC STATUS,0
        GOTO NX@LB127
F1_000494 EQU $ ; IN [MITLEDXP.BAS]
THR[GCTR] = "0"
        MOVF GCTR,W
        ADDLW THR
        MOVWF FSR
        MOVLW 48
        MOVWF INDF
F1_000495 EQU $ ; IN [MITLEDXP.BAS]
TMN[GCTR] = "0"
        MOVF GCTR,W
```

```
        ADDLW TMN
        MOVWF FSR
        MOVLW 48
        MOVWF INDF
F1_000498 EQU $ ; IN [MITLEDXP.BAS]
NEXT GCTR
        INCF GCTR,F
        SET@PAGE FR@LB126
        BTFSS STATUS,2
        GOTO FR@LB126
NX@LB127
F1_000500 EQU $ ; IN [MITLEDXP.BAS]
THR[2] = 0
        CLRF THR#2
F1_000501 EQU $ ; IN [MITLEDXP.BAS]
TMN[2] = 0
        CLRF TMN#2
F1_000503 EQU $ ; IN [MITLEDXP.BAS]
GCTR = 0
        CLRF GCTR
F1_000504 EQU $ ; IN [MITLEDXP.BAS]
KEYTEMP = "0"
        MOVLW 48
        MOVWF KEYTEMP
F1_000506 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 2,1
        MOVLW 128
        MOVWF BPFH
        MOVLW 192
        F@CALL LCD@CRS
F1_000508 EQU $ ; IN [MITLEDXP.BAS]
WHILE 1 = 1
BC@LL127
F1_000510 EQU $ ; IN [MITLEDXP.BAS]
GOSUB MYNUMKEYPAD
        F@CALL MYNUMKEYPAD
F1_000512 EQU $ ; IN [MITLEDXP.BAS]
IF KEYTEMP = "*" THEN
        MOVLW 42
        SUBWF KEYTEMP,W
        SET@PAGE BC@LL130
        BTFSS STATUS,2
        GOTO BC@LL130
F1_000514 EQU $ ; IN [MITLEDXP.BAS]
IF GCTR < 3 THEN
        MOVLW 3
        SUBWF GCTR,W
        SET@PAGE BC@LL132
        BTFSC STATUS,0
        GOTO BC@LL132
F1_000516 EQU $ ; IN [MITLEDXP.BAS]
GCTR = GCTR + 1
        INCF GCTR,F
        F@JUMP BC@LL133
BC@LL132
F1_000518 EQU $ ; IN [MITLEDXP.BAS]
ELSE
F1_000520 EQU $ ; IN [MITLEDXP.BAS]
BREAK
        F@JUMP BC@LL128
F1_000522 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL133
        F@JUMP BC@LL129
F1_000525 EQU $ ; IN [MITLEDXP.BAS]
ELSEIF KEYTEMP = "#" THEN
BC@LL130
        MOVLW 35
        SUBWF KEYTEMP,W
        SET@PAGE BC@LL134
        BTFSS STATUS,2
        GOTO BC@LL134
F1_000527 EQU $ ; IN [MITLEDXP.BAS]
IF GCTR > 0 THEN
        MOVF GCTR,F
        SET@PAGE BC@LL136
        BTFSC STATUS,2
        GOTO BC@LL136
F1_000529 EQU $ ; IN [MITLEDXP.BAS]
GCTR = GCTR - 1
        DECF GCTR,F
F1_000531 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL136
        F@JUMP BC@LL137
BC@LL134
F1_000533 EQU $ ; IN [MITLEDXP.BAS]
ELSE
F1_000535 EQU $ ; IN [MITLEDXP.BAS]
IF GCTR < 2 THEN
        MOVLW 2
        SUBWF GCTR,W
        SET@PAGE BC@LL139
        BTFSC STATUS,0
        GOTO BC@LL139
F1_000537 EQU $ ; IN [MITLEDXP.BAS]
THR[GCTR] = KEYTEMP
        MOVF GCTR,W
        ADDLW THR
        MOVWF FSR
        MOVF KEYTEMP,W
        MOVWF INDF
F1_000538 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 2, GCTR + 1,THR[GCTR]
        MOVLW 128
        MOVWF BPFH
        MOVLW 2
        MOVWF GEN
        INCF GCTR,W
        F@CALL CURSOR
        MOVF GCTR,W
        ADDLW THR
        MOVWF FSR
        MOVF INDF,W
```

```
        F@CALL PRINT
        F@JUMP BC@LL140
BC@LL139
F1_000539 EQU $ ; IN [MITLEDXP.BAS]
ELSE
F1_000541 EQU $ ; IN [MITLEDXP.BAS]
TMN[GCTR - 2] = KEYTEMP
        MOVLW 2
        SUBWF GCTR,W
        ADDLW TMN
        MOVWF FSR
        MOVF KEYTEMP,W
        MOVWF INDF
F1_000542 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 2, GCTR + 4,TMN[GCTR - 2]
        MOVLW 128
        MOVWF BPFH
        MOVLW 2
        MOVWF GEN
        MOVLW 4
        ADDWF GCTR,W
        F@CALL CURSOR
        MOVLW 2
        SUBWF GCTR,W
        ADDLW TMN
        MOVWF FSR
        MOVF INDF,W
        F@CALL PRINT
F1_000544 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL140
F1_000550 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL137
BC@LL129
F1_000552 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 2, GCTR + 1
        MOVLW 128
        MOVWF BPFH
        MOVLW 2
        MOVWF GEN
        INCF GCTR,W
        F@CALL CURSOR
F1_000554 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL127
BC@LL128
F1_000557 EQU $ ; IN [MITLEDXP.BAS]
AHR = VAL(THR,DEC)
        MOVLW 72
        F@CALL VALDEC@32
        MOVWF AHR
F1_000558 EQU $ ; IN [MITLEDXP.BAS]
AMN = VAL(TMN,DEC)
        MOVLW 75
        F@CALL VALDEC@32
        MOVWF AMN
F1_000561 EQU $ ; IN [MITLEDXP.BAS]
GOSUB MYNUMKEYPAD
        F@CALL MYNUMKEYPAD
F1_000563 EQU $ ; IN [MITLEDXP.BAS]
CLS
        F@CALL LCD@CLS
F1_000564 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 1,1,"SIZE: ", DEC2 AHR, " X
", DEC2 AMN
        MOVLW 128
        MOVWF BPFH
        F@CALL LCD@CRS
        MOVLW HIGH (STR@LB5)
        MOVWF GEN4H
        MOVLW LOW (STR@LB5)
        F@CALL C@STROUT
        MOVLW 2
        MOVWF GEN4H
        MOVF AHR,W
        F@CALL OUT@DECC
        MOVLW 32
        F@CALL PRINT
        MOVLW 120
        F@CALL PRINT
        MOVLW 32
        F@CALL PRINT
        MOVLW 2
        MOVWF GEN4H
        MOVF AMN,W
        F@CALL OUT@DECC
F1_000565 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 2,1,"*-OK  | #-CANCEL"
        MOVLW 128
        MOVWF BPFH
        MOVLW 192
        F@CALL LCD@CRS
        MOVLW HIGH (STR@LB6)
        MOVWF GEN4H
        MOVLW LOW (STR@LB6)
        F@CALL C@STROUT
F1_000568 EQU $ ; IN [MITLEDXP.BAS]
WHILE 1 = 1
BC@LL141
F1_000570 EQU $ ; IN [MITLEDXP.BAS]
GOSUB MYNUMKEYPAD
        F@CALL MYNUMKEYPAD
F1_000572 EQU $ ; IN [MITLEDXP.BAS]
IF KEYTEMP = "*" THEN
        MOVLW 42
        SUBWF KEYTEMP,W
        SET@PAGE BC@LL144
        BTFSS STATUS,2
        GOTO BC@LL144
F1_000574 EQU $ ; IN [MITLEDXP.BAS]
IF AMN > 20 OR AMN < 1 THEN
        MOVF AMN,W
        MOVWF PP0
```

```
        MOVLW 20
        F@CALL T@GTB
        MOVWF SP#P9
        MOVLW 1
        SUBWF AMN,W
        MOVLW 1
        BTFSC STATUS,0
        CLRW
        IORWF SP#P9,F
        SET@PAGE BC@LL146
        BTFSC STATUS,2
        GOTO BC@LL146
F1_000576 EQU $ ; IN [MITLEDXP.BAS]
CLS
        F@CALL LCD@CLS
F1_000577 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 1,1,"INVALID SIZE"
        MOVLW 128
        MOVWF BPFH
        F@CALL LCD@CRS
        MOVLW HIGH (STR@LB7)
        MOVWF GEN4H
        MOVLW LOW (STR@LB7)
        F@CALL C@STROUT
F1_000578 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000579 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 1464
BC@LL147
        MOVLW 5
        SUBWF DSDELAYH,W
        MOVLW 184
        BTFSC STATUS,2
        SUBWF DSDELAY,W
        SET@PAGE BC@LL148
        BTFSC STATUS,0
        GOTO BC@LL148
F1_000580 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL147
BC@LL148
F1_000582 EQU $ ; IN [MITLEDXP.BAS]
GOTO STSTART
        F@JUMP STSTART
        F@JUMP BC@LL145
F1_000584 EQU $ ; IN [MITLEDXP.BAS]
ELSEIF AHR> 20 OR AHR < 1 THEN
BC@LL146
        MOVF AHR,W
        MOVWF PP0
        MOVLW 20
        F@CALL T@GTB
        MOVWF SP#P9
        MOVLW 1
        SUBWF AHR,W
        MOVLW 1
        BTFSC STATUS,0
        CLRW
        IORWF SP#P9,F
        SET@PAGE BC@LL149
        BTFSC STATUS,2
        GOTO BC@LL149
F1_000586 EQU $ ; IN [MITLEDXP.BAS]
CLS
        F@CALL LCD@CLS
F1_000587 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 1,1,"INVALID SIZE"
        MOVLW 128
        MOVWF BPFH
        F@CALL LCD@CRS
        MOVLW HIGH (STR@LB8)
        MOVWF GEN4H
        MOVLW LOW (STR@LB8)
        F@CALL C@STROUT
F1_000588 EQU $ ; IN [MITLEDXP.BAS]
DSDELAY = 0
        CLRF DSDELAY
        CLRF DSDELAYH
F1_000589 EQU $ ; IN [MITLEDXP.BAS]
WHILE DSDELAY < 1464
BC@LL150
        MOVLW 5
        SUBWF DSDELAYH,W
        MOVLW 184
        BTFSC STATUS,2
        SUBWF DSDELAY,W
        SET@PAGE BC@LL151
        BTFSC STATUS,0
        GOTO BC@LL151
F1_000590 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL150
BC@LL151
F1_000592 EQU $ ; IN [MITLEDXP.BAS]
GOTO STSTART
        F@JUMP STSTART
        F@JUMP BC@LL152
BC@LL149
F1_000594 EQU $ ; IN [MITLEDXP.BAS]
ELSE
F1_000596 EQU $ ; IN [MITLEDXP.BAS]
GOTO STOK
        F@JUMP STOK
F1_000598 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL152
BC@LL145
        F@JUMP BC@LL143
F1_000601 EQU $ ; IN [MITLEDXP.BAS]
ELSEIF KEYTEMP = "#" THEN
BC@LL144
        MOVLW 35
```

```
        SUBWF KEYTEMP,W
        SET@PAGE BC@LL153
        BTFSS STATUS,2
        GOTO BC@LL153
F1_000603 EQU $ ; IN [MITLEDXP.BAS]
GOTO STSTART
        F@JUMP STSTART
F1_000605 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL153
BC@LL143
F1_000608 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL141
BC@LL142
STOK
F1_000612 EQU $ ; IN [MITLEDXP.BAS]
CLS
        F@CALL LCD@CLS
F1_000613 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 1,1,"1.ACETATE"
        MOVLW 128
        MOVWF BPFH
        F@CALL LCD@CRS
        MOVLW HIGH (STR@LB9)
        MOVWF GEN4H
        MOVLW LOW (STR@LB9)
        F@CALL C@STROUT
F1_000614 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 2,1,"2.T. PAPER"
        MOVLW 128
        MOVWF BPFH
        MOVLW 192
        F@CALL LCD@CRS
        MOVLW HIGH (STR@LB10)
        MOVWF GEN4H
        MOVLW LOW (STR@LB10)
        F@CALL C@STROUT
F1_000616 EQU $ ; IN [MITLEDXP.BAS]
WHILE 1 = 1
BC@LL154
F1_000618 EQU $ ; IN [MITLEDXP.BAS]
GOSUB MYNUMKEYPAD
        F@CALL MYNUMKEYPAD
F1_000620 EQU $ ; IN [MITLEDXP.BAS]
IF KEYTEMP = "1" OR KEYTEMP = "2"
THEN
        MOVLW 49
        SUBWF KEYTEMP,W
        MOVLW 1
        BTFSS STATUS,2
        MOVLW 0
        MOVWF SP#P9
        MOVLW 50
        SUBWF KEYTEMP,W
        MOVLW 1
        BTFSS STATUS,2
        MOVLW 0
        IORWF SP#P9,F
        SET@PAGE BC@LL157
        BTFSC STATUS,2
        GOTO BC@LL157
F1_000622 EQU $ ; IN [MITLEDXP.BAS]
ATYPE = KEYTEMP - $30
        MOVLW 48
        SUBWF KEYTEMP,W
        MOVWF ATYPE
F1_000623 EQU $ ; IN [MITLEDXP.BAS]
BREAK
        F@JUMP BC@LL155
F1_000625 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL157
F1_000627 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL154
BC@LL155
F1_000629 EQU $ ; IN [MITLEDXP.BAS]
CLS
        F@CALL LCD@CLS
F1_000630 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 1,1,"1.SINGLE-SIDED"
        MOVLW 128
        MOVWF BPFH
        F@CALL LCD@CRS
        MOVLW HIGH (STR@LB11)
        MOVWF GEN4H
        MOVLW LOW (STR@LB11)
        F@CALL C@STROUT
F1_000631 EQU $ ; IN [MITLEDXP.BAS]
PRINT AT 2,1,"2.DOUBLE-SIDED"
        MOVLW 128
        MOVWF BPFH
        MOVLW 192
        F@CALL LCD@CRS
        MOVLW HIGH (STR@LB12)
        MOVWF GEN4H
        MOVLW LOW (STR@LB12)
        F@CALL C@STROUT
F1_000633 EQU $ ; IN [MITLEDXP.BAS]
WHILE 1 = 1
BC@LL158
F1_000635 EQU $ ; IN [MITLEDXP.BAS]
GOSUB MYNUMKEYPAD
        F@CALL MYNUMKEYPAD
F1_000637 EQU $ ; IN [MITLEDXP.BAS]
IF KEYTEMP = "1" OR KEYTEMP = "2"
THEN
        MOVLW 49
        SUBWF KEYTEMP,W
        MOVLW 1
        BTFSS STATUS,2
        MOVLW 0
        MOVWF SP#P9
```

```
        MOVLW 50
        SUBWF KEYTEMP,W
        MOVLW 1
        BTFSS STATUS,2
        MOVLW 0
        IORWF SP#P9,F
        SET@PAGE BC@LL161
        BTFSC STATUS,2
        GOTO BC@LL161
F1_000639 EQU $ ; IN [MITLEDXP.BAS]
DSMODE = KEYTEMP - $30
        MOVLW 48
        SUBWF KEYTEMP,W
        MOVWF DSMODE
F1_000640 EQU $ ; IN [MITLEDXP.BAS]
BREAK
        F@JUMP BC@LL159
F1_000642 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL161
F1_000644 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL158
BC@LL159
F1_000646 EQU $ ; IN [MITLEDXP.BAS]
GIE = 0
        BCF INTCON,7
F1_000648 EQU $ ; IN [MITLEDXP.BAS]
MN = 0
        CLRF MN
F1_000649 EQU $ ; IN [MITLEDXP.BAS]
SC = 0
        CLRF SC
F1_000653 EQU $ ; IN [MITLEDXP.BAS]
SELECT CASE AHR
F1_000655 EQU $ ; IN [MITLEDXP.BAS]
CASE 1
        MOVLW 1
        SUBWF AHR,W
        SET@PAGE BC@LL163
        BTFSS STATUS,2
        GOTO BC@LL163
F1_000657 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $01
        MOVLW 1
        MOVWF PORTB
F1_000659 EQU $ ; IN [MITLEDXP.BAS]
IF DSMODE = 2 THEN
        MOVLW 2
        SUBWF DSMODE,W
        SET@PAGE BC@LL166
        BTFSS STATUS,2
        GOTO BC@LL166
F1_000661 EQU $ ; IN [MITLEDXP.BAS]
PORTB0_S = 1
        BSF PORTD,2
F1_000662 EQU $ ; IN [MITLEDXP.BAS]
PORTB1_S = 0
        BCF PORTD,3
F1_000663 EQU $ ; IN [MITLEDXP.BAS]
PORTB2_S = 0
        BCF PORTD,4
F1_000664 EQU $ ; IN [MITLEDXP.BAS]
PORTB3_S = 0
        BCF PORTC,4
F1_000665 EQU $ ; IN [MITLEDXP.BAS]
PORTB4_S = 0
        BCF PORTC,5
F1_000666 EQU $ ; IN [MITLEDXP.BAS]
PORTB5_S = 0
        BCF PORTC,6
F1_000667 EQU $ ; IN [MITLEDXP.BAS]
PORTB6_S = 0
        BCF PORTC,7
F1_000668 EQU $ ; IN [MITLEDXP.BAS]
PORTB7_S = 0
        BCF PORTA,5
F1_000670 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL166
        F@JUMP BC@LL162
BC@LL163
F1_000673 EQU $ ; IN [MITLEDXP.BAS]
CASE 2
        MOVLW 2
        SUBWF AHR,W
        SET@PAGE BC@LL168
        BTFSS STATUS,2
        GOTO BC@LL168
F1_000675 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $03
        MOVLW 3
        MOVWF PORTB
F1_000677 EQU $ ; IN [MITLEDXP.BAS]
IF DSMODE = 2 THEN
        MOVLW 2
        SUBWF DSMODE,W
        SET@PAGE BC@LL170
        BTFSS STATUS,2
        GOTO BC@LL170
F1_000679 EQU $ ; IN [MITLEDXP.BAS]
PORTB0_S = 1
        BSF PORTD,2
F1_000680 EQU $ ; IN [MITLEDXP.BAS]
PORTB1_S = 1
        BSF PORTD,3
F1_000681 EQU $ ; IN [MITLEDXP.BAS]
PORTB2_S = 0
        BCF PORTD,4
F1_000682 EQU $ ; IN [MITLEDXP.BAS]
PORTB3_S = 0
        BCF PORTC,4
```

```
F1_000683 EQU $ ; IN [MITLEDXP.BAS]
PORTB4_S = 0
        BCF PORTC,5
F1_000684 EQU $ ; IN [MITLEDXP.BAS]
PORTB5_S = 0
        BCF PORTC,6
F1_000685 EQU $ ; IN [MITLEDXP.BAS]
PORTB6_S = 0
        BCF PORTC,7
F1_000686 EQU $ ; IN [MITLEDXP.BAS]
PORTB7_S = 0
        BCF PORTA,5
F1_000688 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL170
        F@JUMP BC@LL162
BC@LL168
F1_000691 EQU $ ; IN [MITLEDXP.BAS]
CASE 3, 4
        MOVLW 3
        SUBWF AHR,W
        SET@PAGE BC@LL171
        BTFSC STATUS,2
        GOTO BC@LL171
        MOVLW 4
        SUBWF AHR,W
        SET@PAGE BC@LL172
        BTFSS STATUS,2
        GOTO BC@LL172
BC@LL171
F1_000693 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $07
        MOVLW 7
        MOVWF PORTB
F1_000695 EQU $ ; IN [MITLEDXP.BAS]
IF DSMODE = 2 THEN
        MOVLW 2
        SUBWF DSMODE,W
        SET@PAGE BC@LL174
        BTFSS STATUS,2
        GOTO BC@LL174
F1_000697 EQU $ ; IN [MITLEDXP.BAS]
PORTB0_S = 1
        BSF PORTD,2
F1_000698 EQU $ ; IN [MITLEDXP.BAS]
PORTB1_S = 1
        BSF PORTD,3
F1_000699 EQU $ ; IN [MITLEDXP.BAS]
PORTB2_S = 1
        BSF PORTD,4
F1_000700 EQU $ ; IN [MITLEDXP.BAS]
PORTB3_S = 0
        BCF PORTC,4
F1_000701 EQU $ ; IN [MITLEDXP.BAS]
PORTB4_S = 0
        BCF PORTC,5
F1_000702 EQU $ ; IN [MITLEDXP.BAS]
PORTB5_S = 0
        BCF PORTC,6
F1_000703 EQU $ ; IN [MITLEDXP.BAS]
PORTB6_S = 0
        BCF PORTC,7
F1_000704 EQU $ ; IN [MITLEDXP.BAS]
PORTB7_S = 0
        BCF PORTA,5
F1_000706 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL174
        F@JUMP BC@LL162
BC@LL172
F1_000709 EQU $ ; IN [MITLEDXP.BAS]
CASE 5
        MOVLW 5
        SUBWF AHR,W
        SET@PAGE BC@LL176
        BTFSS STATUS,2
        GOTO BC@LL176
F1_000711 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $0F
        MOVLW 15
        MOVWF PORTB
F1_000713 EQU $ ; IN [MITLEDXP.BAS]
IF DSMODE = 2 THEN
        MOVLW 2
        SUBWF DSMODE,W
        SET@PAGE BC@LL178
        BTFSS STATUS,2
        GOTO BC@LL178
F1_000715 EQU $ ; IN [MITLEDXP.BAS]
PORTB0_S = 1
        BSF PORTD,2
F1_000716 EQU $ ; IN [MITLEDXP.BAS]
PORTB1_S = 1
        BSF PORTD,3
F1_000717 EQU $ ; IN [MITLEDXP.BAS]
PORTB2_S = 1
        BSF PORTD,4
F1_000718 EQU $ ; IN [MITLEDXP.BAS]
PORTB3_S = 1
        BSF PORTC,4
F1_000719 EQU $ ; IN [MITLEDXP.BAS]
PORTB4_S = 0
        BCF PORTC,5
F1_000720 EQU $ ; IN [MITLEDXP.BAS]
PORTB5_S = 0
        BCF PORTC,6
F1_000721 EQU $ ; IN [MITLEDXP.BAS]
PORTB6_S = 0
        BCF PORTC,7
F1_000722 EQU $ ; IN [MITLEDXP.BAS]
PORTB7_S = 0
        BCF PORTA,5
```

```
F1_000724 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL178
        F@JUMP BC@LL162
BC@LL176
F1_000727 EQU $ ; IN [MITLEDXP.BAS]
CASE 6, 7
        MOVLW 6
        SUBWF AHR,W
        SET@PAGE BC@LL179
        BTFSC STATUS,2
        GOTO BC@LL179
        MOVLW 7
        SUBWF AHR,W
        SET@PAGE BC@LL180
        BTFSS STATUS,2
        GOTO BC@LL180
BC@LL179
F1_000729 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $1F
        MOVLW 31
        MOVWF PORTB
F1_000731 EQU $ ; IN [MITLEDXP.BAS]
IF DSMODE = 2 THEN
        MOVLW 2
        SUBWF DSMODE,W
        SET@PAGE BC@LL182
        BTFSS STATUS,2
        GOTO BC@LL182
F1_000733 EQU $ ; IN [MITLEDXP.BAS]
PORTB0_S = 1
        BSF PORTD,2
F1_000734 EQU $ ; IN [MITLEDXP.BAS]
PORTB1_S = 1
        BSF PORTD,3
F1_000735 EQU $ ; IN [MITLEDXP.BAS]
PORTB2_S = 1
        BSF PORTD,4
F1_000736 EQU $ ; IN [MITLEDXP.BAS]
PORTB3_S = 1
        BSF PORTC,4
F1_000737 EQU $ ; IN [MITLEDXP.BAS]
PORTB4_S = 1
        BSF PORTC,5
F1_000738 EQU $ ; IN [MITLEDXP.BAS]
PORTB5_S = 0
        BCF PORTC,6
F1_000739 EQU $ ; IN [MITLEDXP.BAS]
PORTB6_S = 0
        BCF PORTC,7
F1_000740 EQU $ ; IN [MITLEDXP.BAS]
PORTB7_S = 0
        BCF PORTA,5
F1_000742 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL182
        F@JUMP BC@LL162
BC@LL180
F1_000745 EQU $ ; IN [MITLEDXP.BAS]
CASE 8, 9
        MOVLW 8
        SUBWF AHR,W
        SET@PAGE BC@LL183
        BTFSC STATUS,2
        GOTO BC@LL183
        MOVLW 9
        SUBWF AHR,W
        SET@PAGE BC@LL184
        BTFSS STATUS,2
        GOTO BC@LL184
BC@LL183
F1_000747 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $3F
        MOVLW 63
        MOVWF PORTB
F1_000749 EQU $ ; IN [MITLEDXP.BAS]
IF DSMODE = 2 THEN
        MOVLW 2
        SUBWF DSMODE,W
        SET@PAGE BC@LL186
        BTFSS STATUS,2
        GOTO BC@LL186
F1_000751 EQU $ ; IN [MITLEDXP.BAS]
PORTB0_S = 1
        BSF PORTD,2
F1_000752 EQU $ ; IN [MITLEDXP.BAS]
PORTB1_S = 1
        BSF PORTD,3
F1_000753 EQU $ ; IN [MITLEDXP.BAS]
PORTB2_S = 1
        BSF PORTD,4
F1_000754 EQU $ ; IN [MITLEDXP.BAS]
PORTB3_S = 1
        BSF PORTC,4
F1_000755 EQU $ ; IN [MITLEDXP.BAS]
PORTB4_S = 1
        BSF PORTC,5
F1_000756 EQU $ ; IN [MITLEDXP.BAS]
PORTB5_S = 1
        BSF PORTC,6
F1_000757 EQU $ ; IN [MITLEDXP.BAS]
PORTB6_S = 0
        BCF PORTC,7
F1_000758 EQU $ ; IN [MITLEDXP.BAS]
PORTB7_S = 0
        BCF PORTA,5
F1_000760 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL186
        F@JUMP BC@LL162
BC@LL184
F1_000763 EQU $ ; IN [MITLEDXP.BAS]
CASE 10
        MOVLW 10
```

```
        SUBWF AHR,W
        SET@PAGE BC@LL188
        BTFSS STATUS,2
        GOTO BC@LL188
F1_000765 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $7F
        MOVLW 127
        MOVWF PORTB
F1_000767 EQU $ ; IN [MITLEDXP.BAS]
IF DSMODE = 2 THEN
        MOVLW 2
        SUBWF DSMODE,W
        SET@PAGE BC@LL190
        BTFSS STATUS,2
        GOTO BC@LL190
F1_000769 EQU $ ; IN [MITLEDXP.BAS]
PORTB0_S = 1
        BSF PORTD,2
F1_000770 EQU $ ; IN [MITLEDXP.BAS]
PORTB1_S = 1
        BSF PORTD,3
F1_000771 EQU $ ; IN [MITLEDXP.BAS]
PORTB2_S = 1
        BSF PORTD,4
F1_000772 EQU $ ; IN [MITLEDXP.BAS]
PORTB3_S = 1
        BSF PORTC,4
F1_000773 EQU $ ; IN [MITLEDXP.BAS]
PORTB4_S = 1
        BSF PORTC,5
F1_000774 EQU $ ; IN [MITLEDXP.BAS]
PORTB5_S = 1
        BSF PORTC,6
F1_000775 EQU $ ; IN [MITLEDXP.BAS]
PORTB6_S = 1
        BSF PORTC,7
F1_000776 EQU $ ; IN [MITLEDXP.BAS]
PORTB7_S = 0
        BCF PORTA,5
F1_000778 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL190
        F@JUMP BC@LL162
BC@LL188
F1_000781 EQU $ ; IN [MITLEDXP.BAS]
CASE 11, 12
        MOVLW 11
        SUBWF AHR,W
        SET@PAGE BC@LL191
        BTFSC STATUS,2
        GOTO BC@LL191
        MOVLW 12
        SUBWF AHR,W
        SET@PAGE BC@LL192
        BTFSS STATUS,2
        GOTO BC@LL192
BC@LL191
F1_000783 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $FF
        MOVLW 255
        MOVWF PORTB
F1_000785 EQU $ ; IN [MITLEDXP.BAS]
IF DSMODE = 2 THEN
        MOVLW 2
        SUBWF DSMODE,W
        SET@PAGE BC@LL194
        BTFSS STATUS,2
        GOTO BC@LL194
F1_000787 EQU $ ; IN [MITLEDXP.BAS]
PORTB0_S = 1
        BSF PORTD,2
F1_000788 EQU $ ; IN [MITLEDXP.BAS]
PORTB1_S = 1
        BSF PORTD,3
F1_000789 EQU $ ; IN [MITLEDXP.BAS]
PORTB2_S = 1
        BSF PORTD,4
F1_000790 EQU $ ; IN [MITLEDXP.BAS]
PORTB3_S = 1
        BSF PORTC,4
F1_000791 EQU $ ; IN [MITLEDXP.BAS]
PORTB4_S = 1
        BSF PORTC,5
F1_000792 EQU $ ; IN [MITLEDXP.BAS]
PORTB5_S = 1
        BSF PORTC,6
F1_000793 EQU $ ; IN [MITLEDXP.BAS]
PORTB6_S = 1
        BSF PORTC,7
F1_000794 EQU $ ; IN [MITLEDXP.BAS]
PORTB7_S = 1
        BSF PORTA,5
F1_000796 EQU $ ; IN [MITLEDXP.BAS]
ENDIF
BC@LL194
F1_000800 EQU $ ; IN [MITLEDXP.BAS]
ENDSELECT
BC@LL192
BC@LL162
F1_000802 EQU $ ; IN [MITLEDXP.BAS]
SELECT CASE AMN
F1_000804 EQU $ ; IN [MITLEDXP.BAS]
CASE 1
        MOVLW 1
        SUBWF AMN,W
        SET@PAGE BC@LL196
        BTFSS STATUS,2
        GOTO BC@LL196
F1_000806 EQU $ ; IN [MITLEDXP.BAS]
GSHFL = 0
        CLRF GSHFL
        F@JUMP BC@LL195
BC@LL196
```

```
F1_000808 EQU $ ; IN [MITLEDXP.BAS]
CASE 2
        MOVLW 2
        SUBWF AMN,W
        SET@PAGE BC@LL199
        BTFSS STATUS,2
        GOTO BC@LL199
F1_000811 EQU $ ; IN [MITLEDXP.BAS]
GSHFL = 1
        MOVLW 1
        MOVWF GSHFL
        F@JUMP BC@LL195
BC@LL199
F1_000813 EQU $ ; IN [MITLEDXP.BAS]
CASE 3, 4
        MOVLW 3
        SUBWF AMN,W
        SET@PAGE BC@LL200
        BTFSC STATUS,2
        GOTO BC@LL200
        MOVLW 4
        SUBWF AMN,W
        SET@PAGE BC@LL201
        BTFSS STATUS,2
        GOTO BC@LL201
BC@LL200
F1_000816 EQU $ ; IN [MITLEDXP.BAS]
GSHFL = 2
        MOVLW 2
        MOVWF GSHFL
        F@JUMP BC@LL195
BC@LL201
F1_000818 EQU $ ; IN [MITLEDXP.BAS]
CASE 5
        MOVLW 5
        SUBWF AMN,W
        SET@PAGE BC@LL203
        BTFSS STATUS,2
        GOTO BC@LL203
F1_000821 EQU $ ; IN [MITLEDXP.BAS]
GSHFL = 3
        MOVLW 3
        MOVWF GSHFL
        F@JUMP BC@LL195
BC@LL203
F1_000823 EQU $ ; IN [MITLEDXP.BAS]
CASE 6, 7
        MOVLW 6
        SUBWF AMN,W
        SET@PAGE BC@LL204
        BTFSC STATUS,2
        GOTO BC@LL204
        MOVLW 7
        SUBWF AMN,W
        SET@PAGE BC@LL205
        BTFSS STATUS,2
        GOTO BC@LL205
BC@LL204
F1_000825 EQU $ ; IN [MITLEDXP.BAS]
GSHFL = 4
        MOVLW 4
        MOVWF GSHFL
        F@JUMP BC@LL195
BC@LL205
F1_000827 EQU $ ; IN [MITLEDXP.BAS]
CASE 8, 9
        MOVLW 8
        SUBWF AMN,W
        SET@PAGE BC@LL206
        BTFSC STATUS,2
        GOTO BC@LL206
        MOVLW 9
        SUBWF AMN,W
        SET@PAGE BC@LL207
        BTFSS STATUS,2
        GOTO BC@LL207
BC@LL206
F1_000829 EQU $ ; IN [MITLEDXP.BAS]
GSHFL = 5
        MOVLW 5
        MOVWF GSHFL
        F@JUMP BC@LL195
BC@LL207
F1_000831 EQU $ ; IN [MITLEDXP.BAS]
CASE 10
        MOVLW 10
        SUBWF AMN,W
        SET@PAGE BC@LL209
        BTFSS STATUS,2
        GOTO BC@LL209
F1_000833 EQU $ ; IN [MITLEDXP.BAS]
GSHFL = 6
        MOVLW 6
        MOVWF GSHFL
        F@JUMP BC@LL195
BC@LL209
F1_000835 EQU $ ; IN [MITLEDXP.BAS]
CASE 11, 12
        MOVLW 11
        SUBWF AMN,W
        SET@PAGE BC@LL210
        BTFSC STATUS,2
        GOTO BC@LL210
        MOVLW 12
        SUBWF AMN,W
        SET@PAGE BC@LL211
        BTFSS STATUS,2
        GOTO BC@LL211
BC@LL210
F1_000837 EQU $ ; IN [MITLEDXP.BAS]
GSHFL = 7
        MOVLW 7
        MOVWF GSHFL
```

```
F1_000839 EQU $ ; IN [MITLEDXP.BAS]
ENDSELECT
BC@LL211
BC@LL195
F1_000841 EQU $ ; IN [MITLEDXP.BAS]
GSHFC = 0
        CLRF GSHFC
F1_000843 EQU $ ; IN [MITLEDXP.BAS]
GSHF = 0
        CLRF GSHF
F1_000846 EQU $ ; IN [MITLEDXP.BAS]
MN = ATYPE * 5
        BCF STATUS,0
        RLF ATYPE,W
        ADDWF ATYPE,W
        ADDWF ATYPE,W
        ADDWF ATYPE,W
        MOVWF MN
F1_000847 EQU $ ; IN [MITLEDXP.BAS]
SC = 0
        CLRF SC
F1_000848 EQU $ ; IN [MITLEDXP.BAS]
ICTR = 0
        CLRF ICTRHHH
        CLRF ICTRHH
        CLRF ICTRH
        CLRF ICTR
F1_000853 EQU $ ; IN [MITLEDXP.BAS]
TMR0 = 0
        CLRF TMR0
F1_000854 EQU $ ; IN [MITLEDXP.BAS]
T0IF = 0
        BCF INTCON,2
F1_000855 EQU $ ; IN [MITLEDXP.BAS]
GIE = 1
        BSF INTCON,7
F1_000859 EQU $ ; IN [MITLEDXP.BAS]
RETURN
        RETURN
HWDIAGNOSTICS
F1_000863 EQU $ ; IN [MITLEDXP.BAS]
PORTB = $FF
        MOVLW 255
        MOVWF PORTB
F1_000864 EQU $ ; IN [MITLEDXP.BAS]
PORTB0_S = 1
        BSF PORTD,2
F1_000865 EQU $ ; IN [MITLEDXP.BAS]
PORTB1_S = 1
        BSF PORTD,3
F1_000866 EQU $ ; IN [MITLEDXP.BAS]
PORTB2_S = 1
        BSF PORTD,4
F1_000867 EQU $ ; IN [MITLEDXP.BAS]
PORTB3_S = 1
        BSF PORTC,4
F1_000868 EQU $ ; IN [MITLEDXP.BAS]
PORTB4_S = 1
        BSF PORTC,5
F1_000869 EQU $ ; IN [MITLEDXP.BAS]
PORTB5_S = 1
        BSF PORTC,6
F1_000870 EQU $ ; IN [MITLEDXP.BAS]
PORTB6_S = 1
        BSF PORTC,7
F1_000871 EQU $ ; IN [MITLEDXP.BAS]
PORTB7_S = 1
        BSF PORTA,5
F1_000873 EQU $ ; IN [MITLEDXP.BAS]
WHILE 1 = 1
BC@LL212
F1_000877 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD & $1F
        MOVLW 31
        ANDWF PORTD,F
F1_000878 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD | $00
F1_000879 EQU $ ; IN [MITLEDXP.BAS]
DELAYMS 1000
        MOVLW 3
        MOVWF PP1H
        MOVLW 232
        F@CALL DLY@W
F1_000881 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD & $1F
        MOVLW 31
        ANDWF PORTD,F
F1_000882 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD | $20
        BSF PORTD,5
F1_000883 EQU $ ; IN [MITLEDXP.BAS]
DELAYMS 1000
        MOVLW 3
        MOVWF PP1H
        MOVLW 232
        F@CALL DLY@W
F1_000885 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD & $1F
        MOVLW 31
        ANDWF PORTD,F
F1_000886 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD | $40
        BSF PORTD,6
F1_000887 EQU $ ; IN [MITLEDXP.BAS]
DELAYMS 1000
        MOVLW 3
        MOVWF PP1H
        MOVLW 232
        F@CALL DLY@W
F1_000889 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD & $1F
        MOVLW 31
        ANDWF PORTD,F
```

```
F1_000890 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD | $60
        MOVLW 96
        IORWF PORTD,F
F1_000891 EQU $ ; IN [MITLEDXP.BAS]
DELAYMS 1000
        MOVLW 3
        MOVWF PP1H
        MOVLW 232
        F@CALL DLY@W
F1_000893 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD & $1F
        MOVLW 31
        ANDWF PORTD,F
F1_000894 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD | $80
        BSF PORTD,7
F1_000895 EQU $ ; IN [MITLEDXP.BAS]
DELAYMS 1000
        MOVLW 3
        MOVWF PP1H
        MOVLW 232
        F@CALL DLY@W
F1_000897 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD & $1F
        MOVLW 31
        ANDWF PORTD,F
F1_000898 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD | $A0
        MOVLW 160
        IORWF PORTD,F
F1_000899 EQU $ ; IN [MITLEDXP.BAS]
DELAYMS 1000
        MOVLW 3
        MOVWF PP1H
        MOVLW 232
        F@CALL DLY@W
F1_000901 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD & $1F
        MOVLW 31
        ANDWF PORTD,F
F1_000902 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD | $C0
        MOVLW 192
        IORWF PORTD,F
F1_000903 EQU $ ; IN [MITLEDXP.BAS]
DELAYMS 1000
        MOVLW 3
        MOVWF PP1H
        MOVLW 232
        F@CALL DLY@W
F1_000905 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD & $1F
        MOVLW 31
        ANDWF PORTD,F
F1_000906 EQU $ ; IN [MITLEDXP.BAS]
PORTD = PORTD | $E0
        MOVLW 224
        IORWF PORTD,F
F1_000907 EQU $ ; IN [MITLEDXP.BAS]
DELAYMS 1000
        MOVLW 3
        MOVWF PP1H
        MOVLW 232
        F@CALL DLY@W
F1_000911 EQU $ ; IN [MITLEDXP.BAS]
WEND
        F@JUMP BC@LL212
BC@LL213
F1_000914 EQU $ ; IN [MITLEDXP.BAS]
RETURN
        RETURN
F1_000916 EQU $ ; IN [MITLEDXP.BAS]
END
PB@LB215
        SLEEP
        F@JUMP PB@LB215
F1_EOF EQU $ ; MITLEDXP.BAS
PB@LB216
        F@JUMP PB@LB216
STR@LB1
        DE 69,88,80,79,83,85,82,69,0
STR@LB2
        DE
67,79,77,80,76,69,84,69,68,46,0
STR@LB3
        DE
83,69,84,32,80,67,66,32,83,73,90,69,
0
STR@LB4
        DE
48,48,32,120,32,48,48,32,105,110,99,
104,0
STR@LB5
        DE 83,73,90,69,58,32,0
STR@LB6
        DE
42,45,79,75,32,32,124,32,35,45,67,65
,78,67,69,76,0
STR@LB7
        DE
73,78,86,65,76,73,68,32,83,73,90,69,
0
STR@LB8
        DE
73,78,86,65,76,73,68,32,83,73,90,69,
0
STR@LB9
        DE
49,46,65,99,101,116,97,116,101,0
STR@LB10
        DE
50,46,84,46,32,80,97,112,101,114,0
STR@LB11
```

```
        DE 49,46,83,105,110,103,108,101,45,83,105,100,101,100,0
STR@LB12
        DE 50,46,68,111,117,98,108,101,45,83,105,100,101,100,0
__EOF
__CONFIG HS_OSC & WDT_OFF & PWRTE_ON & BODEN_OFF & LVP_OFF & CP_OFF &
CPD_OFF & DEBUG_OFF
        END
```

APPENDIX D

DATA SHEETS

PIC16F87XA
Data Sheet

28/40/44-Pin Enhanced
Flash Microcontrollers

MICROCHIP

PIC16F87XA

28/40/44-Pin Enhanced Flash Microcontrollers

Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory, Up to 368 x 8 bytes of Data Memory (RAM), Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers

Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I^2C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) - 8 bits wide with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (V_{REF}) module
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs are externally accessible

Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

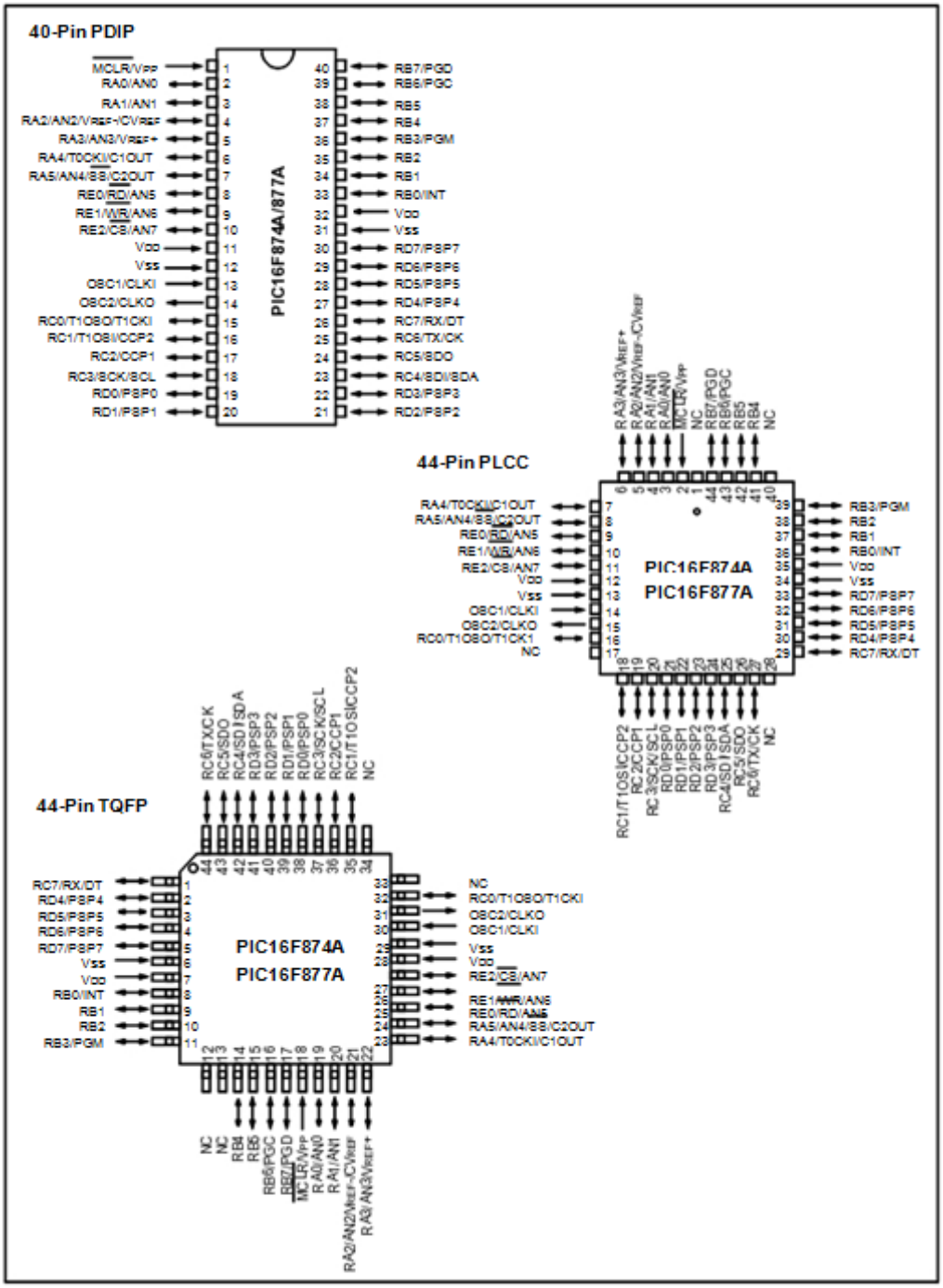
CMOS Technology:

- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

Device	Program Memory		Data SRAM	EEPROM	I/O	10-bit	CCP	MSSP		USART	Timers	Comparators
	Bytes	# Single Word Instructions	(Bytes)	(Bytes)		A/D (ch)	(PWM)	SPI	Master I²C		8/16-bit	
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

© 2003 Microchip Technology Inc.

Pin Diagrams (Continued)



1.0 DEVICE OVERVIEW

This document contains device specific information about the following devices:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

PIC16F873A/876A devices are available only in 28-pin packages, while PIC16F874A/877A devices are available in 40-pin and 44-pin packages. All devices in the PIC16F87XA family share common architecture with the following differences:

- The PIC16F873A and PIC16F874A have one-half of the total on-chip memory of the PIC16F876A and PIC16F877A
- The 28-pin devices have three I/O ports, while the 40/44-pin devices have five
- The 28-pin devices have fourteen interrupts, while the 40/44-pin devices have fifteen
- The 28-pin devices have five A/D input channels, while the 40/44-pin devices have eight
- The Parallel Slave Port is implemented only on the 40/44-pin devices

The available features are summarized in Table 1-1. Block diagrams of the PIC16F873A/876A and PIC16F874A/877A devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2 and Table 1-3.

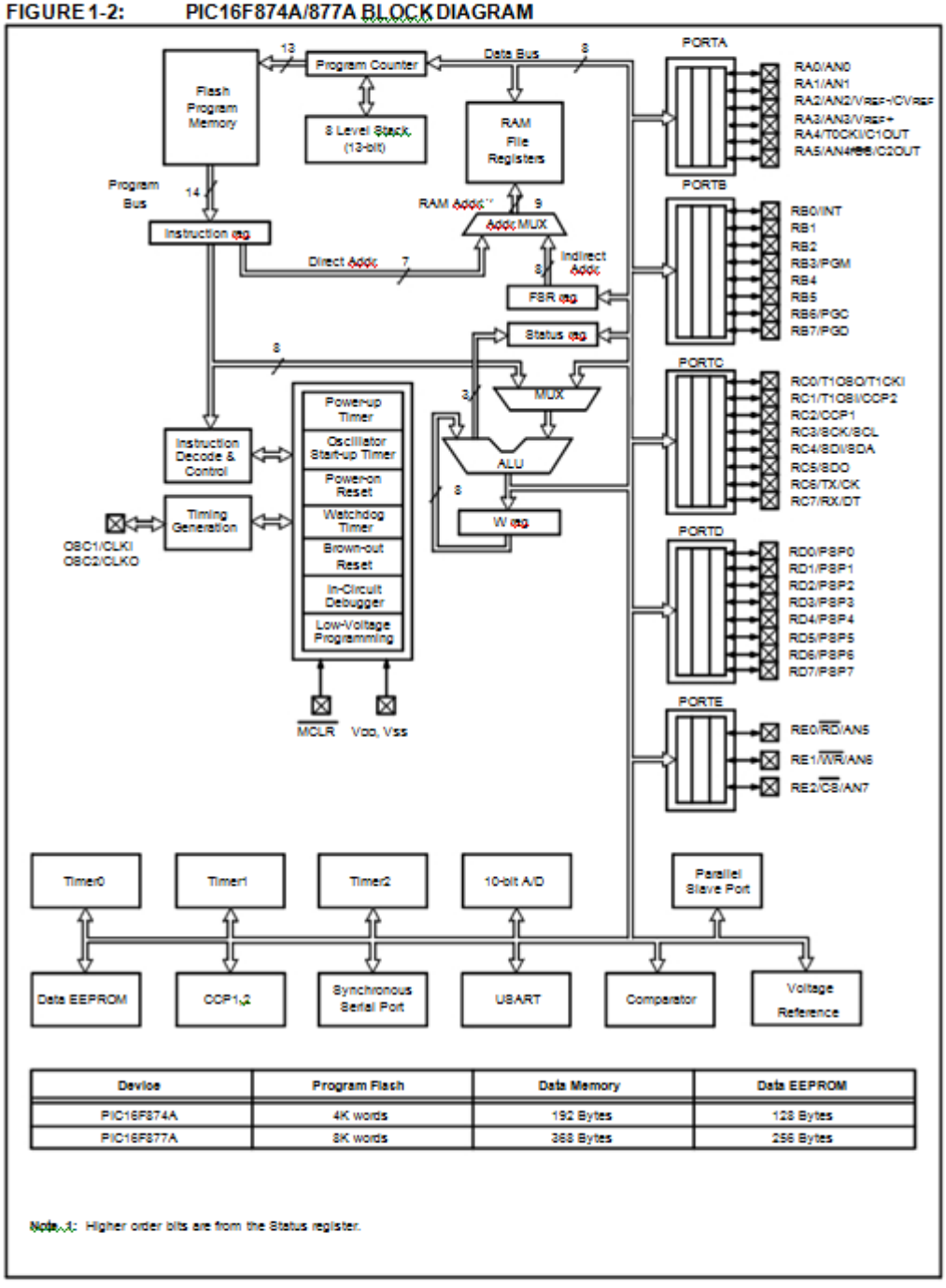
Additional information may be found in the PICmicro® Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

TABLE 1-1: PIC16F87XA DEVICE FEATURES

Key Features	PIC16F873A	PIC16F874A	PIC16F876A	PIC16F877A
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
Flash Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory (bytes)	128	128	256	256
Interrupts	14	15	14	15
I/O Ports	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C	Ports A, B, C, D, E
Timers	3	3	3	3
Capture/Compare/PWM modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Analog Comparators	2	2	2	2
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions
Packages	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN

© 2003 Microchip Technology Inc.

FIGURE 1-2: PIC16F874A/877A BLOCK DIAGRAM



Device	Program Flash	Data Memory	Data EEPROM
PIC16F874A	4K words	192 Bytes	128 Bytes
PIC16F877A	8K words	368 Bytes	256 Bytes

Note 1: Higher order bits are from the Status register.

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKI	13	14	30	32		ST/CMOS(4)	Oscillator crystal or external clock input.
OSC1					I		Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise, CMOS.
CLKI					I		External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins).
OSC2/CLKO	14	15	31	33		—	Oscillator crystal or clock output.
OSC2					O		Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO					O		In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
MCLR/VPP	1	2	18	18		ST	Master Clear (input) or programming voltage (output).
MCLR					I		Master Clear (Reset) input. This pin is an active low Reset to the device.
VPP					P		Programming voltage input.
							PORTA is a bidirectional I/O port.
RA0/AN0	2	3	19	19		TTL	
RA0					I/O		Digital I/O.
AN0					I		Analog input 0.
RA1/AN1	3	4	20	20		TTL	
RA1					I/O		Digital I/O.
AN1					I		Analog input 1.
RA2/AN2/VREF-/CVREF	4	5	21	21		TTL	
RA2					I/O		Digital I/O.
AN2					I		Analog input 2.
VREF-					I		A/D reference voltage (Low) input.
CVREF					O		Comparator VREF output.
RA3/AN3/VREF+	5	6	22	22		TTL	
RA3					I/O		Digital I/O.
AN3					I		Analog input 3.
VREF+					I		A/D reference voltage (High) input.
RA4/T0CKI/C1OUT	6	7	23	23		ST	
RA4					I/O		Digital I/O - Open-drain when configured as output.
T0CKI					I		Timer0 external clock input.
C1OUT					O		Comparator 1 output.
RA5/AN4/SS/C2OUT	7	8	24	24		TTL	
RA5					I/O		Digital I/O.
AN4					I		Analog input 4.
SS					I		SPI slave select input.
C2OUT					O		Comparator 2 output.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

© 2003 Microchip Technology Inc.

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
							PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	33	36	8	9		TTL/ST[1]	
RB0					I/O		Digital I/O.
INT					I		External interrupt.
RB1	34	37	9	10	I/O	TTL	Digital I/O.
RB2	35	38	10	11	I/O	TTL	Digital I/O.
RB3/PGM	36	39	11	12		TTL	
RB3					I/O		Digital I/O.
PGM					I		Low-voltage ICSP programming enable pin.
RB4	37	41	14	14	I/O	TTL	Digital I/O.
RB5	38	42	15	15	I/O	TTL	Digital I/O.
RB6/PGC	39	43	16	16		TTL/ST[2]	
RB6					I/O		Digital I/O.
PGC					I		In-circuit debugger and ICSP programming clock.
RB7/PGD	40	44	17	17		TTL/ST[2]	
RB7					I/O		Digital I/O.
PGD					I/O		In-circuit debugger and ICSP programming data.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

© 2003 Microchip Technology Inc.

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
							PORTC is a bidirectional I/O port.
RC0/T1OSO/T1CKI	15	16	32	34		ST	
RC0					I/O		Digital I/O.
T1OSO					O		Timer1 oscillator output.
T1CKI					I		Timer1 external clock input.
RC1/T1OSI/CCP2	16	18	35	35		ST	
RC1					I/O		Digital I/O.
T1OSI					I		Timer1 oscillator input.
CCP2					I/O		Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1	17	19	36	36		ST	
RC2					I/O		Digital I/O.
CCP1					I/O		Capture1 input, Compare1 output, PWM1 output.
RC3/SCK/SCL	18	20	37	37		ST	
RC3					I/O		Digital I/O.
SCK					I/O		Synchronous serial clock input/output for SPI mode.
SCL					I/O		Synchronous serial clock input/output for I^2C mode.
RC4/SDI/SDA	23	25	42	42		ST	
RC4					I/O		Digital I/O.
SDI					I		SPI data in.
SDA					I/O		I2C data I/O.
RC5/SDO	24	26	43	43		ST	
RC5					I/O		Digital I/O.
SDO					O		SPI data out.
RC6/TX/CK	25	27	44	44		ST	
RC6					I/O		Digital I/O.
TX					O		USART asynchronous transmit.
CK					I/O		USART1 synchronous clock.
RC7/RX/DT	26	29	1	1		ST	
RC7					I/O		Digital I/O.
RX					I		USART asynchronous receive.
DT					I/O		USART synchronous data.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

© 2003 Microchip Technology Inc.

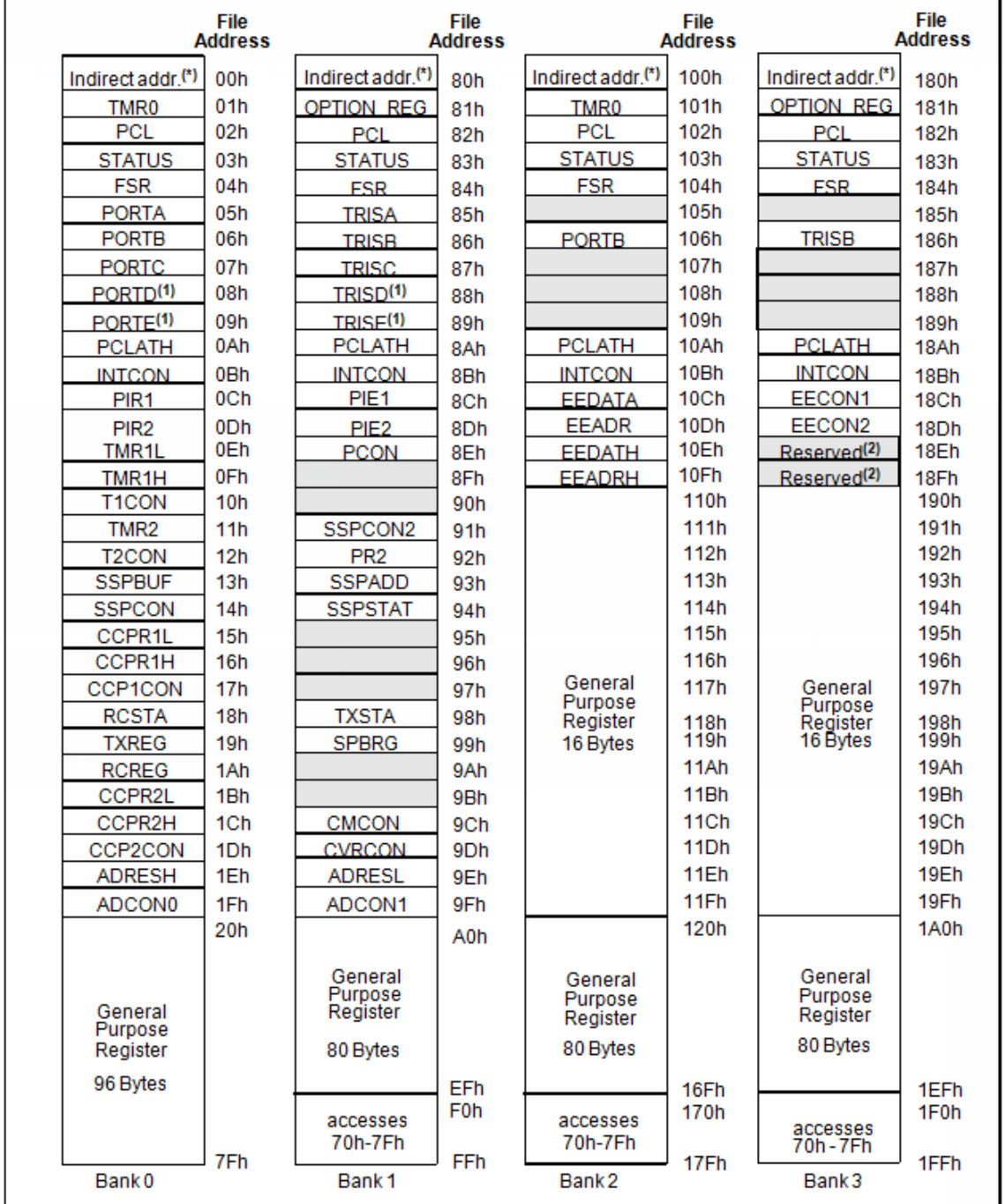
TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
							PORTD is a bidirectional I/O port or Parallel Slave Port when interfacing to a microprocessor bus.
RD0/PSP0	19	21	38	38		ST/TTL(3)	
RD0					I/O		Digital I/O.
PSP0					I/O		Parallel Slave Port data.
RD1/PSP1	20	22	39	39		ST/TTL(3)	
RD1					I/O		Digital I/O.
PSP1					I/O		Parallel Slave Port data.
RD2/PSP2	21	23	40	40		ST/TTL(3)	
RD2					I/O		Digital I/O.
PSP2					I/O		Parallel Slave Port data.
RD3/PSP3	22	24	41	41		ST/TTL(3)	
RD3					I/O		Digital I/O.
PSP3					I/O		Parallel Slave Port data.
RD4/PSP4	27	30	2	2		ST/TTL(3)	
RD4					I/O		Digital I/O.
PSP4					I/O		Parallel Slave Port data.
RD5/PSP5	28	31	3	3		ST/TTL(3)	
RD5					I/O		Digital I/O.
PSP5					I/O		Parallel Slave Port data.
RD6/PSP6	29	32	4	4		ST/TTL(3)	
RD6					I/O		Digital I/O.
PSP6					I/O		Parallel Slave Port data.
RD7/PSP7	30	33	5	5		ST/TTL(3)	
RD7					I/O		Digital I/O.
PSP7					I/O		Parallel Slave Port data.
							PORTE is a bidirectional I/O port.
RE0/$\overline{RD}$/AN5	8	9	25	25		ST/TTL(3)	
RE0					I/O		Digital I/O.
$\overline{RD}$					I		Read control for Parallel Slave Port.
AN5					I		Analog input 5.
RE1/$\overline{WR}$/AN6	9	10	26	26		ST/TTL(3)	
RE1					I/O		Digital I/O.
$\overline{WR}$					I		Write control for Parallel Slave Port.
AN6					I		Analog input 6.
RE2/$\overline{CS}$/AN7	10	11	27	27		ST/TTL(3)	
RE2					I/O		Digital I/O.
$\overline{CS}$					I		Chip select control for Parallel Slave Port.
AN7					I		Analog input 7.
VSS	12, 31	13, 34	6, 29	6, 30, 31	P	—	Ground reference for logic and I/O pins.
VDD	11, 32	12, 35	7, 28	7, 8, 28, 29	P	—	Positive supply for logic and I/O pins.
NC	—	1, 17, 28, 40	12,13, 33, 34	13	—	—	These pins are not internally connected. These pins should be left unconnected.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

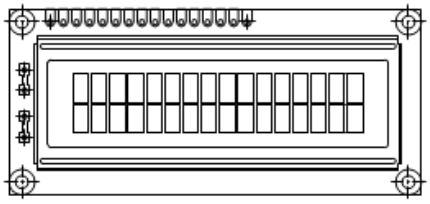
FIGURE 2-3: PIC16F876A/877A REGISTER FILE MAP



Unimplemented data memory locations, read as '0'. * Not a physical register.

Note 1: These registers are not implemented on the PIC16F876A.
2: These registers are reserved; maintain these registers clear.

16 x 2 Character LCD



FEATURES

- 5 x 8 dots with cursor
- Built-in controller (KS 0066 or Equivalent)
- + 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)
- N.V. optional for + 3V power supply

MECHANICAL DATA

ITEM	STANDARD VALUE	UNIT
Module Dimension	80.0 x 36.0	mm
Viewing Area	66.0 x 16.0	mm
Dot Size	0.56 x 0.66	mm
Character Size	2.96 x 5.56	mm

ABSOLUTE MAXIMUM RATING

ITEM	SYMBOL	STANDARD VALUE			UNIT
		MIN.	TYP.	MAX.	
Power Supply	VDD-VSS	- 0.3	-	7.0	V
Input Voltage	VI	- 0.3	-	VDD	V

NOTE: VSS = 0 Volt, VDD = 5.0 Volt

ELECTRICAL SPECIFICATIONS

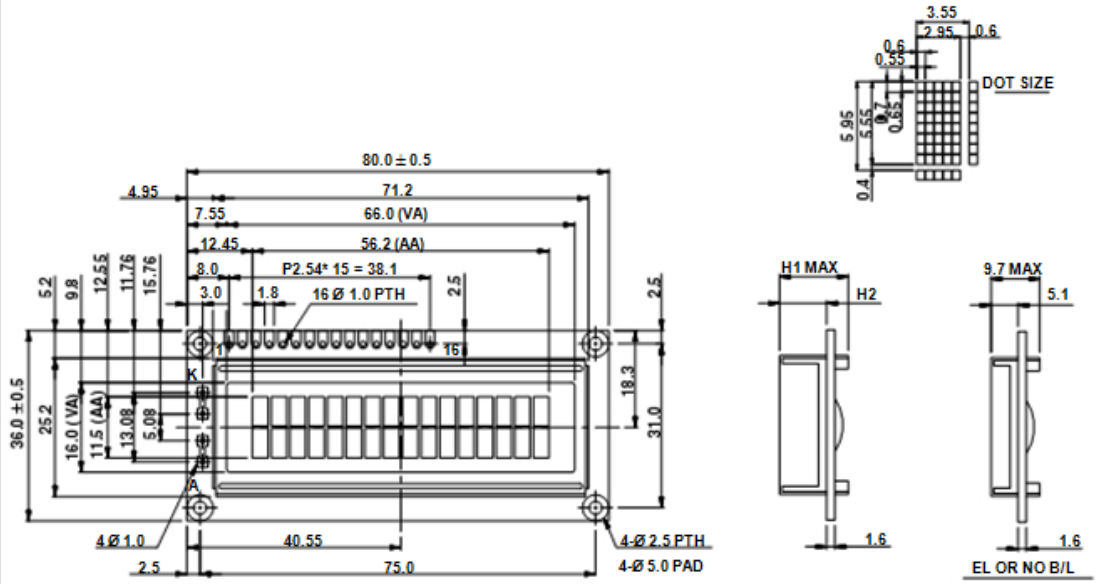
ITEM	SYMBOL	CONDITION		STANDARD VALUE			UNIT
				MIN.	TYP.	MAX.	
Input Voltage	VDD	VDD = + 5V		4.7	5.0	5.3	V
		VDD = + 3V		2.7	3.0	5.3	V
Supply Current	IDD	VDD = 5V		-	1.2	3.0	mA
Recommended LC Driving Voltage for Normal Temp. Version Module	VDD - V0	- 20 °C		-	-	-	V
		0°C		4.2	4.8	5.1	
		25°C		3.8	4.2	4.6	
		50°C		3.6	4.0	4.4	
		70°C		-	-	-	
LED Forward Voltage	VF	25°C		-	4.2	4.6	V
LED Forward Current	IF	25°C	Array	-	130	260	mA
			Edge	-	20	40	
EL Power Supply Current	IEL	Vel = 110VAC:400Hz		-	-	5.0	mA

DISPLAY CHARACTER ADDRESS CODE:

Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01														0F
DD RAM Address	40	41														4F

PIN NUMBER	SYMBOL	FUNCTION
1	Vss	GND
2	Vdd	+3V or +5V
3	Vo	Contrast Adjustment
4	RS	H/L Register Select Signal
5	R/$\overline{W}$	H/L Read/Write Signal
6	E	H → L Enable Signal
7	DB0	H/L Data Bus Line
8	DB1	H/L Data Bus Line
9	DB2	H/L Data Bus Line
10	DB3	H/L Data Bus Line
11	DB4	H/L Data Bus Line
12	DB5	H/L Data Bus Line
13	DB6	H/L Data Bus Line
14	DB7	H/L Data Bus Line
15	A/Vee	+4.2V for LED/Negative Voltage Output
16	K	Power Supply for B/L (0V)

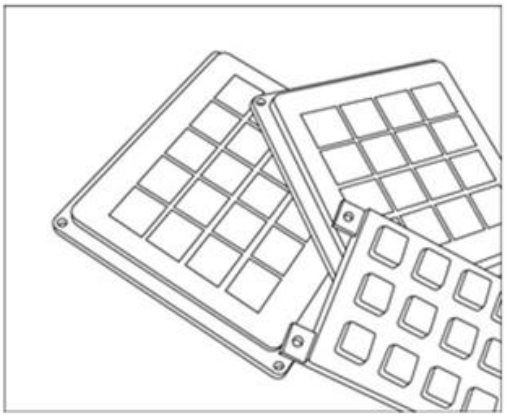
DIMENSIONS in millimeters



LED - H/L B/L	HIGH	LOW
H1	13.2	12.1
H2	8.6	7.5

KEYPAD

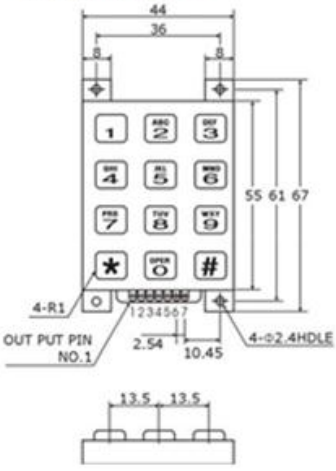
Keypad Switch



3 x 4 KEYPAD

– Pin Layout

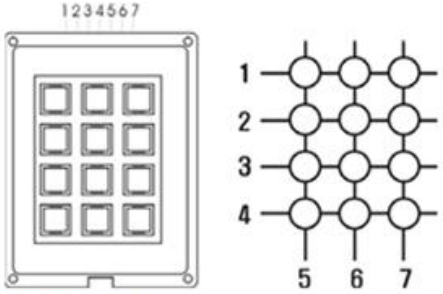
– Dimensions (Units : mm)



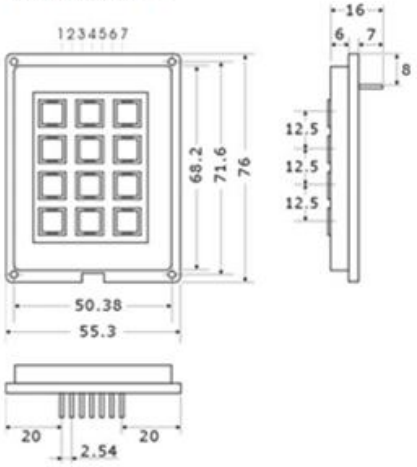
3 x 4 FUNCTION KEYPAD –

Customize Keys

– Pin Layout



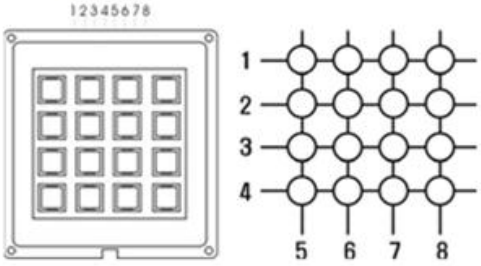
– Dimensions (Units : mm)



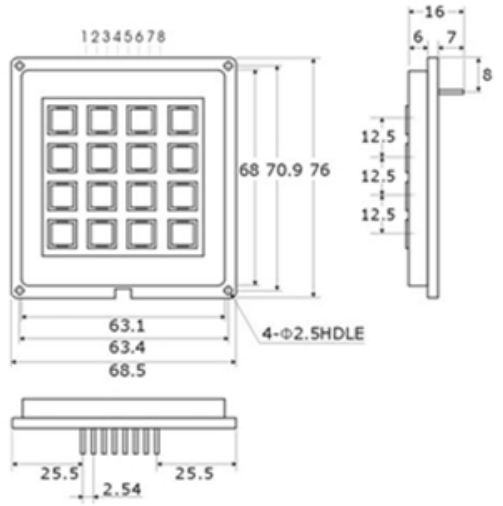
4 x 4 KEYPAD

– Customize Keys

– Pin Layout



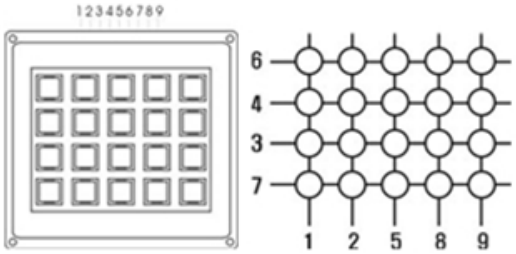
– Dimensions (Unit : mm)



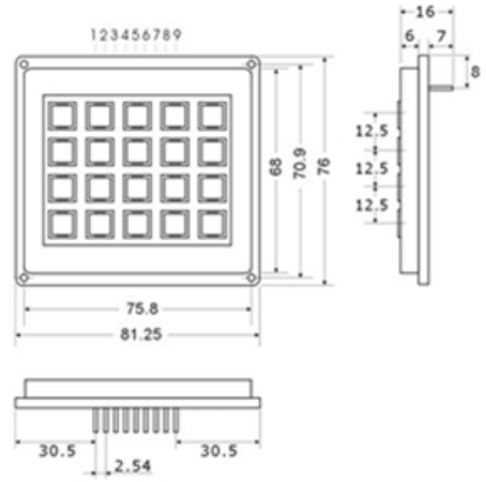
▌4 x 5 KEYPAD

– Customized Keys

– Pin Layout



– Dimensions (Units : mm)

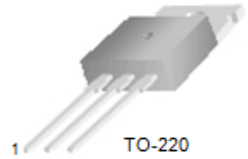


FAIRCHILD SEMICONDUCTOR™

TIP32 Series(TIP32/32A/32B/32C)

Medium Power Linear Switching Applications

- Complement to TIP31/31A/31B/31C



1 TO-220

1.Base 2.Collector 3.Emitter

PNP Epitaxial Silicon Transistor

Absolute Maximum Ratings T_C=25°C unless otherwise noted

Symbol	Parameter	Value	Units
V_{CBO}	Collector-Base Voltage : TIP32	- 40	V
	: TIP32A	- 60	V
	: TIP32B	- 80	V
	: TIP32C	- 100	V
V_{CEO}	Collector-Emitter Voltage : TIP32	- 40	V
	: TIP32A	- 60	V
	: TIP32B	- 80	V
	: TIP32C	-100	V
V_{EBO}	Emitter-Base Voltage	- 5	V
I_C	Collector Current (DC)	- 3	A
I_{CP}	Collector Current (Pulse)	- 5	A
I_B	Base Current	- 3	A
P_C	Collector Dissipation (T_C=25°C)	40	W
P_C	Collector Dissipation (T_a=25°C)	2	W
T_J	Junction Temperature	150	°C
T_{STG}	Storage Temperature	- 65 ~ 150	°C

Electrical Characteristics T_C=25°C unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Max.	Units
V_{CEO}(sus)	* Collector-Emitter Sustaining Voltage : TIP32	I_C = - 30mA, I_B = 0	-40		V
	: TIP32A		-60		V
	: TIP32B		-80		V
	: TIP32C		-100		V
I_{CEO}	Collector Cut-off Current : TIP32/32A	V_{CE} = - 30V, I_B = 0		- 0.3	mA
	: TIP32B/32C	V_{CE} = - 60V, I_B = 0		- 0.3	mA
I_{CES}	Collector Cut-off Current : TIP32	V_{CE} = - 40V, V_{EB} = 0		- 200	µA
	: TIP32A	V_{CE} = - 60V, V_{EB} = 0		- 200	µA
	: TIP32B	V_{CE} = - 80V, V_{EB} = 0		- 200	µA
	: TIP32C	V_{CE} = - 100V, V_{CE} = 0		- 200	µA
I_{EBO}	Emitter Cut-off Current	V_{EB} = - 5V, I_C = 0		- 1	mA
h_{FE}	* DC Current Gain	V_{CE} = - 4V, I_C = - 1A	25		
		V_{CE} = - 4V, I_C = - 3A	10	50	
V_{CE}(sat)	* Collector-Emitter Saturation Voltage	I_C = - 3A, I_B = - 375mA		- 1.2	V
V_{BE}(sat)	* Base-Emitter Saturation Voltage	V_{CE} = - 4V, I_C = - 3A		- 1.8	V
f_T	Current Gain Bandwidth Product	V_{CE} = - 10V, I_C = - 500mA	3.0		MHz

* Pulse Test: PW≤300µs, Duty Cycle≤2%

©2000 Fairchild Semiconductor International

Rev. A, February 2000

Typical Characteristics

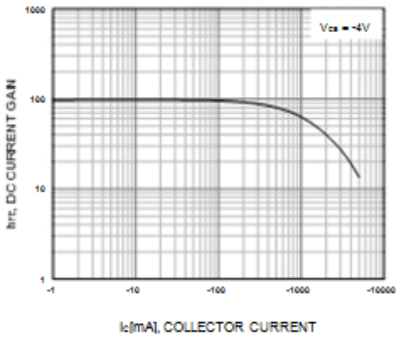


Figure 1. DC current Gain

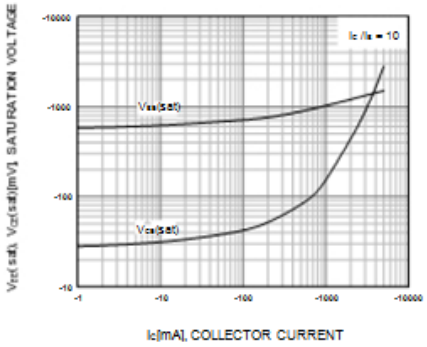


Figure 2. Base-Emitter Saturation Voltage
Collector-Emitter Saturation Voltage

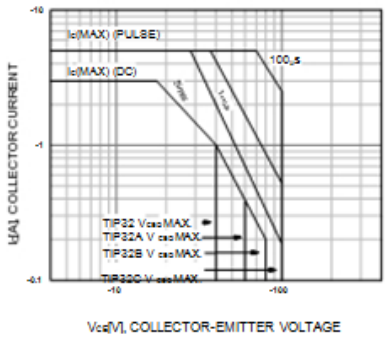


Figure 3. Safe Operating Area

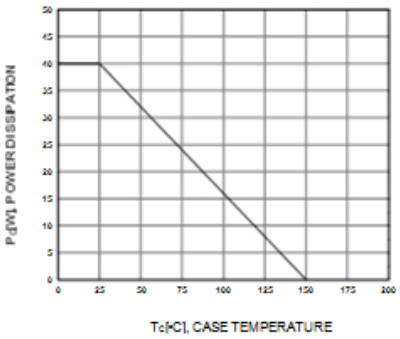
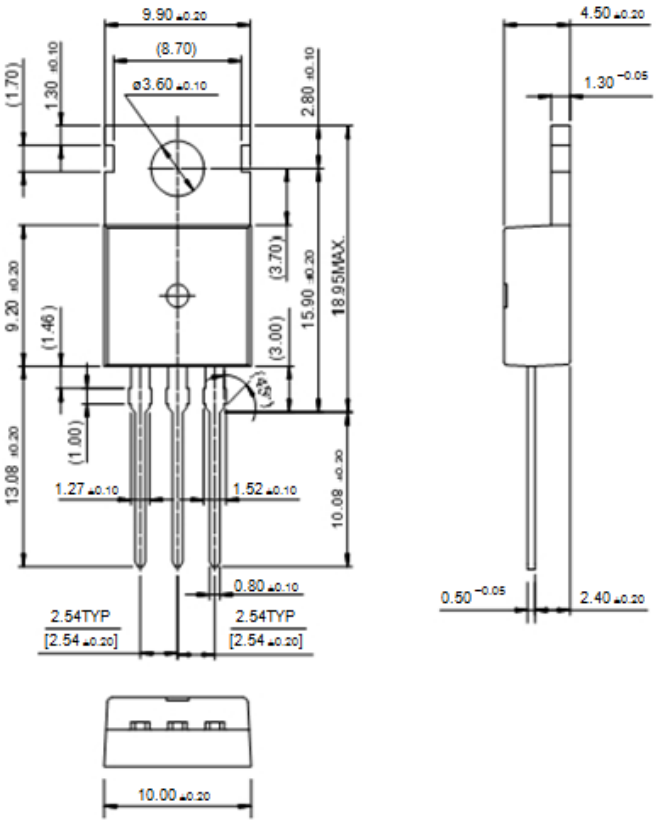


Figure 4. Power Derating

©2000 Fairchild Semiconductor International

Package Demensions

TO-220



Dimensions in Millimeters

©2000 Fairchild Semiconductor International

Silicon NPN Power Transistors 2SC4149

DESCRIPTION

·With ITO-220 package
·Switching power transistor
·Low saturation voltage

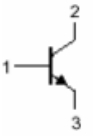


Fig.1 simplified outline (ITO-220) and symbol

PINNING

PIN	DESCRIPTION
1	Base
2	Collector
3	Emitter

Absolute maximum ratings(Ta=25□)

SYMBOL	PARAMETER	CONDITIONS	VALUE	UNIT
V_{CBO}	Collector-base voltage	Open emitter	60	V
V_{CEO}	Collector-emitter voltage	Open base	40	V
V_{EBO}	Emitter-base voltage	Open collector	7	V
I_C	Collector current		10	A
I_{CM}	Collector current-Peak		20	A
I_B	Base current		1.5	A
I_{BM}	Base current-Peak		2	A
P_T	Total power dissipation	T_C=25□	25	W
T_j	Junction temperature		150	□
T_{stg}	Storage temperature		-55~150	□

THERMAL CHARACTERISTICS

SYMBOL	PARAMETER	MAX	UNIT
$R_{th\ j\text{-}C}$	Thermal resistance junction case	5.0	□/W

Silicon NPN Power Transistors 2SC4149

CHARACTERISTICS

Tj=25□ unless otherwise specified

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP.	MAX	UNIT
$V_{CEO(SUS)}$	Collector-emitter sustaining voltage	I_C=0.1A ;I_B=0	40			V
V_{CEsat}	Collector-emitter saturation voltage	I_C=5A; I_B=0.25A			0.3	V
V_{BEsat}	Base-emitter saturation voltage	I_C=5A; I_B=0.25A			1.2	V
I_{CBO}	Collector cut-off current	At rated volatge			0.1	mA
I_{CEO}	Collector cut-off current					
I_{EBO}	Emitter cut-off current	At rated volatge			0.1	mA
h_{FE}	DC current gain	I_C=5A ; V_{CE}=2V	70			
f_T	Transition frequency	I_C=1A ; V_{CE}=10V		50		MHz

Switching times

t_{on}	Turn-on time	I_C=5A;I_{B1}=0.5A I_{B2}=0.5A ,R_L=5Ω V_{BB2}=4V			0.3	μs
t_s	Storage time				1.5	μs
t_f	Fall time				0.5	μs

C106 Series

Preferred Devices

Sensitive Gate Silicon Controlled Rectifiers

Reverse Blocking Thyristors

Glassivated PNPN devices designed for high volume consumer applications such as temperature, light, and speed control; process and remote control, and warning systems where reliability of operation is important.

Features

- Pb−Free Packages are Available*
- Glassivated Surface for Reliability and Uniformity
- Power Rated at Economical Prices
- Practical Level Triggering and Holding Characteristics
- Flat, Rugged, Thermopad Construction for Low Thermal Resistance, High Heat Dissipation and Durability
- Sensitive Gate Triggering
- Device Marking: Device Type, e.g., C106B, Date Code

*For additional information on our Pb−Free strategy and soldering details, please download the ON Semiconductor Soldering and Mounting Techniques Reference Manual, SOLDERRM/D.

ON Semiconductor®

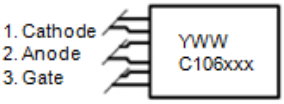
http://onsemi.com

SCRs
4 A RMS, 200 − 600 V

REAR VIEW
SHOW TAB

TO−225AA
CASE 077
STYLE 2

MARKING DIAGRAM & PIN ASSIGNMENT



xx = B, D, D1, D1G*, M, MG*, M1
Y = Year
WW = Work Week
G* = Pb−Free

ORDERING INFORMATION

Device	Package	Shipping†
C106B	TO225AA	500/Box
C106D	TO225AA	500/Box
C106D1**	TO225AA	500/Box
C106D1G	TO225AA (Pb−Free)	500/Box
C106M	TO225AA	500/Box
C106MG	TO225AA (Pb−Free)	500/Box
C106M1**	TO225AA	500/Box

†For information on tape and reel specifications, including part orientation and tape sizes, please refer to our Tape and Reel Packaging Specifications Brochure, BRD8011/D.

**D1 signifies European equivalent for D suffix and M1 signifies European equivalent for M suffix.

Preferred devices are recommended choices for future use and best overall value.

© Semiconductor Components Industries, LLC, 2004

MAXIMUM RATINGS (T_J = 25°C unless otherwise noted)

Characteristic	Symbol	Max	Unit
Peak Repetitive Off−State Voltage (Note 1) (Sine Wave, 50−60 Hz, R_{GK} = 1 kW, T_C = −40° to 110°C)	V_{DRM}, V_{RRM}		V
C106B		200	
C106D, C106D1*		400	
C106M, C106M1*		600	
On-State RMS Current (180° Conduction Angles, T_C = 80°C)	$I_{T(RMS)}$	4.0	A
Average On−State Current (180° Conduction Angles, T_C = 80°C)	$I_{T(AV)}$	2.55	A
Peak Non-Repetitive Surge Current (1/2 Cycle, Sine Wave, 60 Hz, T_J = +110°C)	I_{TSM}	20	A
Circuit Fusing Considerations (t = 8.3 ms)	I^2t	1.65	A^2s
Forward Peak Gate Power (Pulse Width v1.0 msec, T_C = 80°C)	P_{GM}	0.5	W
Forward Average Gate Power (Pulse Width v1.0 msec, T_C = 80°C)	$P_{G(AV)}$	0.1	W
Forward Peak Gate Current (Pulse Width v1.0 msec, T_C = 80°C)	I_{GM}	0.2	A
Operating Junction Temperature Range	T_J	−40 to +110	°C
Storage Temperature Range	T_{stg}	−40 to +150	°C
Mounting Torque (Note 2)	−	6.0	in. lb.

Maximum ratings are those values beyond which device damage can occur. Maximum ratings applied to the device are individual stress limit values (not normal operating conditions) and are not valid simultaneously. If these limits are exceeded, device functional operation is not implied, damage may occur and reliability may be affected.

*D1 signifies European equivalent for D suffix and M1 signifies European equivalent for M suffix.

1. V_{DRM} and V_{RRM} for all types can be applied on a continuous basis. Ratings apply for zero or negative gate voltage; however, positive gate voltage shall not be applied concurrent with negative potential on the anode. Blocking voltages shall not be tested with a constant current source such that the voltage ratings of the devices are exceeded.
2. Torque rating applies with use of compression washer (B52200F006). Mounting torque in excess of 6 in. lb. does not appreciably lower case-to-sink thermal resistance. Anode lead and heatsink contact pad are common.

THERMAL CHARACTERISTICS (T_C = 25°C unless otherwise noted.)

Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction to Case	$R_{\theta JC}$	3.0	°C/W
Thermal Resistance, Junction to Ambient	$R_{\theta JA}$	75	°C/W
Maximum Lead Temperature for Soldering Purposes 1/8" from Case for 10 Seconds	T_L	260	°C

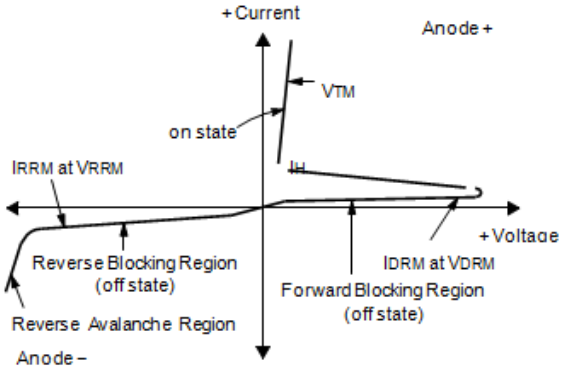
ELECTRICAL CHARACTERISTICS (T_C = 25°C unless otherwise noted.)

Characteristic		Symbol	Min	Typ	Max	Unit
OFF CHARACTERISTICS						
Peak Repetitive Forward or Reverse Blocking Current (V_{AK} = Rated V_{DRM} or V_{RRM}, R_{GK} = 1000 Ohms)	T_J = 25°C T_J = 110°C	I_{DRM}, I_{RRM}	– –	– –	10 100	mA mA
ON CHARACTERISTICS						
Peak Forward On−State Voltage (Note 3) (I_{TM} = 4 A)		V_{TM}	–	–	2.2	Volts
Gate Trigger Current (Continuous dc) (Note 4) (V_{AK} = 6 Vdc, R_L = 100 Ohms)	T_J = 25°C T_J = −40°C	I_{GT}	– –	15 35	200 500	mA
Peak Reverse Gate Voltage (I_{GR} = 10 mA)		V_{GRM}	–	–	6.0	Volts
Gate Trigger Voltage (Continuous dc) (Note 4) (V_{AK} = 6 Vdc, R_L = 100 Ohms)	T_J = 25°C T_J = −40°C	V_{GT}	0.4 0.5	0.60 0.75	0.8 1.0	Volts
Gate Non−Trigger Voltage (Continuous dc) (Note 4) (V_{AK} = 12 V, R_L = 100 Ohms, T_J = 110°C)		V_{GD}	0.2	–	–	Volts
Latching Current (V_{AK} = 12 V, I_G = 20 mA)	T_J = 25°C T_J = −40°C	I_L	– –	0.20 0.35	5.0 7.0	mA
Holding Current (V_D = 12 Vdc) (Initiating Current = 20 mA, Gate Open)	T_J = 25°C T_J = −40°C T_J = +110°C	I_H	– – –	0.19 0.33 0.07	3.0 6.0 2.0	mA
DYNAMIC CHARACTERISTICS						
Critical Rate−of−Rise of Off−State Voltage (V_{AK} = Rated V_{DRM}, Exponential Waveform, R_{GK} = 1000 Ohms, T_J = 110°C)		dv/dt	–	8.0	–	V/ms

3. Pulse Test: Pulse Width ≤ 2.0 ms, Duty Cycle ≤ 2%.
4. R_{GK} is not included in measurement.

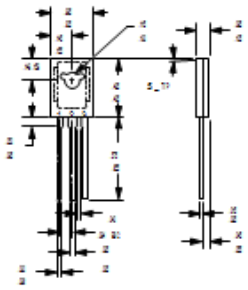
Voltage Current Characteristic of SCR

Symbol	Parameter
V_{DRM}	Peak Repetitive Off State Forward Voltage
I_{DRM}	Peak Forward Blocking Current
V_{RRM}	Peak Repetitive Off State Reverse Voltage
I_{RRM}	Peak Reverse Blocking Current
V_{TM}	Peak On State Voltage
I_H	Holding Current

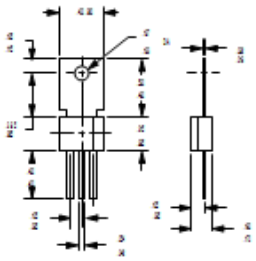


Package Interchangeability

The dimensional diagrams below compare the critical dimensions of the ON Semiconductor C-106 package with competitive devices. It has been demonstrated that the smaller dimensions of the ON Semiconductor package make it compatible in most lead-mount and chassis-mount applications. The user is advised to compare all critical dimensions for mounting compatibility.



ON Semiconductor C-106 Package

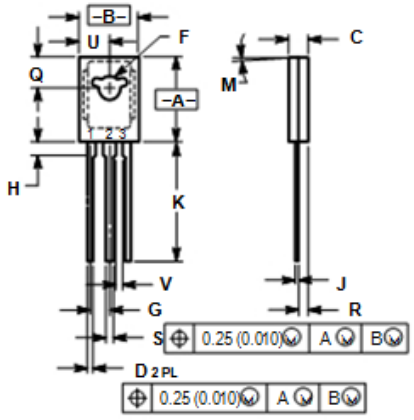


Competitive C-106 Package

C106 Series

PACKAGE DIMENSIONS

TO−225AA
(formerly TO−126)
CASE 077−09
ISSUE Z



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. 077− 01 THRU − 08 OBSOLETE, NEW STANDARD 077− 09.

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.425	0.435	10.80	11.04
B	0.295	0.305	7.50	7.74
C	0.095	0.105	2.42	2.66
D	0.020	0.026	0.51	0.66
F	0.115	0.130	2.93	3.30
G	0.094 BSC		2.39 BSC	
H	0.050	0.095	1.27	2.41
J	0.015	0.025	0.39	0.63
K	0.575	0.655	14.61	16.63
M	5° TYP		5° TYP	
Q	0.148	0.158	3.76	4.01
R	0.045	0.065	1.15	1.65
S	0.025	0.035	0.64	0.88
U	0.145	0.155	3.69	3.93
V	0.040	− − −	1.02	− − −

STYLE 2:
PIN 1. CATHODE
2. ANODE
3. GATE

ON Semiconductor and ON are registered trademarks of Semiconductor Components Industries, LLC (SCILLC). SCILLC reserves the right to make changes without further notice to any products herein. SCILLC makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does SCILLC assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. "Typical" parameters which may be provided in SCILLC data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. SCILLC does not convey any license under its patent rights nor the rights of others. SCILLC products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the SCILLC product could create a situation where personal injury or death may occur. Should Buyer purchase or use SCILLC products for any such unintended or unauthorized application, Buyer shall indemnify and hold SCILLC and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that SCILLC was negligent regarding the design or manufacture of the part. SCILLC is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:
Literature Distribution Center for ON Semiconductor
P.O. Box 61312, Phoenix, Arizona 85082−1312 USA
Phone: 480−829−7710 or 800−344−3860 Toll Free USA/Canada
Fax: 480−829−7709 or 800−344−3867 Toll Free USA/Canada
Email: orderlit@onsemi.com

N. American Technical Support: 800−282−9855 Toll Free USA/Canada

Japan: ON Semiconductor, Japan Customer Focus Center
2−9−1 Kamimeguro, Meguro−ku, Tokyo, Japan 153−0051
Phone: 81−3−5773−3850

ON Semiconductor Website: http://onsemi.com

Order Literature: http://www.onsemi.com/litorder

For additional information, please contact your local Sales Representative.

KA78XX/KA78XXA

3-Terminal 1A Positive Voltage Regulator

Features

- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

Description

The KA78XX/KA78XXA series of three-terminal positive regulator are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.

TO-220

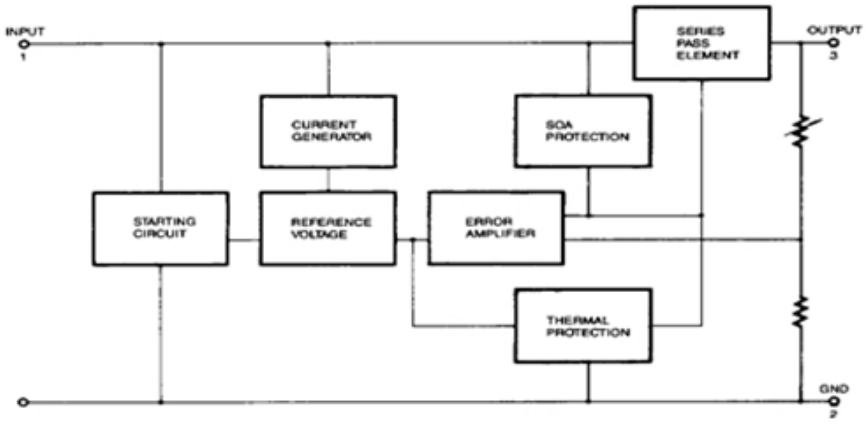
1

D-PAK

1

1. Input 2. GND 3. Output

Internal Block Digram



Rev. 1.0.0

©2001 Fairchild Semiconductor Corporation

Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Input Voltage (for V_O = 5V to 18V) (for V_O = 24V)	V_I V_I	35 40	V V
Thermal Resistance Junction-Cases (TO-220)	$R_{\theta JC}$	5	°C/W
Thermal Resistance Junction-Air (TO-220)	$R_{\theta JA}$	65	°C/W
Operating Temperature Range (KA78XX/A/R)	T_{OPR}	0 ~ +125	°C
Storage Temperature Range	T_{STG}	-65 ~ +150	°C

Electrical Characteristics (KA7805/KA7805R)

(Refer to test circuit ,0°C < T_J < 125°C, I_O = 500mA, V_I =10V, C_I= 0.33µF, C_O=0.1µF, unless otherwise specified)

Parameter	Symbol	Conditions		KA7805			Unit
				Min.	Typ.	Max.	
Output Voltage	V_O	T_J =+25°C		4.8	5.0	5.2	V
		5.0mA ≤ I_O ≤ 1.0A, P_O ≤ 15W V_I = 7V to 20V		4.75	5.0	5.25	
Line Regulation (Note1)	Regline	T_J=+25°C	V_O = 7V to 25V	-	4.0	100	mV
			V_I = 8V to 12V	-	1.6	50	
Load Regulation (Note1)	Regload	T_J=+25°C	I_O = 5.0mA to1.5A	-	9	100	mV
			I_O =250mA to 750mA	-	4	50	
Quiescent Current	I_Q	T_J =+25°C		-	5.0	8.0	mA
Quiescent Current Change	ΔI_Q	I_O = 5mA to 1.0A		-	0.03	0.5	mA
		V_I= 7V to 25V		-	0.3	1.3	
Output Voltage Drift	$\Delta V_O/\Delta T$	I_O= 5mA		-	-0.8	-	mV/°C
Output Noise Voltage	V_N	f = 10Hz to 100KHz, T_A=+25°C		-	42	-	µV/V_O
Ripple Rejection	RR	f = 120Hz V_O = 8V to 18V		62	73	-	dB
Dropout Voltage	V_{Drop}	I_O = 1A, T_J =+25°C		-	2	-	V
Output Resistance	r_O	f = 1KHz		-	15	-	mΩ
Short Circuit Current	I_{SC}	V_I = 35V, T_A =+25°C		-	230	-	mA
Peak Current	I_{PK}	T_J =+25°C		-	2.2	-	A

Note:

1. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.